IBM Copyright Permission #24352

“Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation”

Each reprint must be accompanied by the following credit line: “Reprint Courtesy of International Business Machines Corporation, © International Business Machines Corporation”. **The credit line normally should appear on the page where the reprint/post appears, either under the title or as a footnote.**

If the foregoing is inconvenient, the credit line may be placed on the face or back of the title page (or front cover, if there is no title page) or in a conveniently viewable manner with suitable reference to the place where the reprint/post appears.

When multiple IBM materials are reprinted/posted, a consolidated credit paragraph may be used on the title page, or in a conveniently viewable manner listing the titles, corresponding copyright notices and references to the points where the reprints/posts appear.

It is the understanding of International Business Machines Corporation **that the purpose for which its material is being used is accurate and true as stated in the original request.**

Permission to quote from, transmit electronically or reprint IBM material is limited to the purpose and quantities originally requested and must not be construed as a blanket license to use the material for other purposes or to reprint other IBM copyrighted material.

IBM reserves the right to withdraw permission to use copyrighted material whenever, in its discretion, it feels that the privilege of using its material is being used in a way detrimental to its interest or the above instructions are not being followed properly to protect its copyright.

No permission is granted to use trademarks of International Business Machines Corporation **and its affiliates apart from the incidental appearance of such trademarks in the titles, text, and illustrations of the named publications. Any proposed use of trademarks apart from such incidental appearance requires separate approval in writing and ordinarily cannot be given. The use of any IBM trademark should not be of a manner which might cause confusion of origin or appear to endorse non-IBM products.**

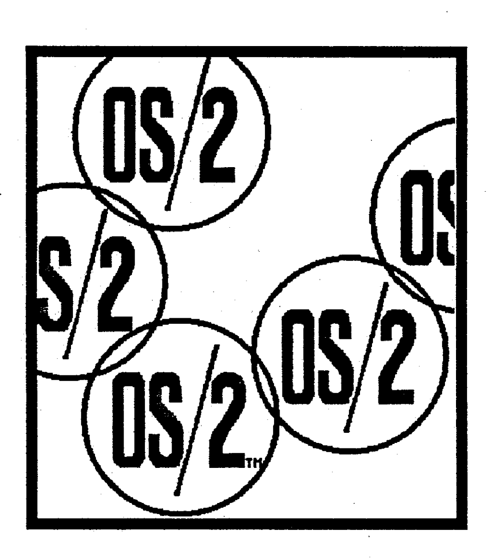
THIS PERMISSION IS PROVIDED WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE**.**

INTERNATIONAL BUSINESS MACHINES CORPORATION

Dated: December 11, 2015

OS/2 Version 2.1
Architektur und Funktionen

Merkblätter



Kurstyp 78J8H

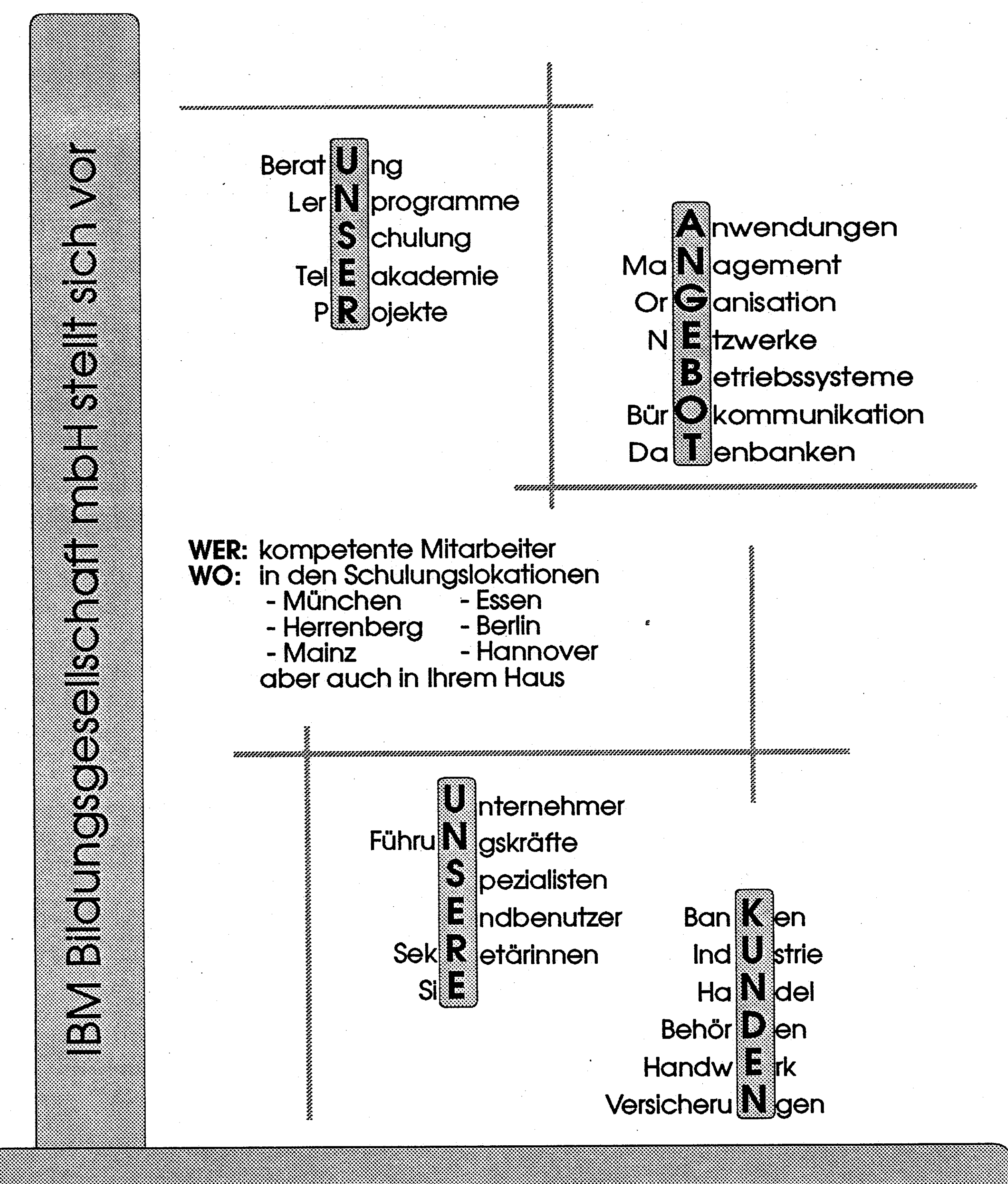
Zur Erstellung dieser Unterlagen wurden folgende IBM Produkte eingesetzt.

Document Composition Facility, Program No. 5748-XX9
Graphical Data Display Manager, Program No. 5748-XXH
IBM Publishing Systems BookMaster, Program No. 5688-015
SCRIPT Mathematical Formular Formatter, Program No. 5798-DPW
IBM Publishing Systems ProcessMaster, Program No. 5666-328
IBM Publishing Systems BrowseMaster, Program No. 5688-009
IBM Publishing Systems DrawMaster, Program No. 5664-388
IBM OS/2 Image Support, Program No. 5688-174
IBM OS/2 Live Parsing Editor, Program No. 5604-286
Image Handling Facility, Program No. 5664-363
Print Service Facility/Virtual Machine, Program No. 5564-198

Dieses Dokument wurde am 21.10.1993 auf dem Drucker IBM 3827 erstellt.

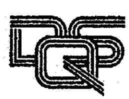
Zuständig für weitere sachliche Auskünfte zum Inhalt dieser Veröffentlichung:

IBM Deutschland Bildungsgesellschaft mbH
BI BZ Nord
Laatzener Straße 1
30539 Hannover



Ihre Fragen beantworten wir gerne unter der Rufnummer 0130-5000

Zertifiziert nach DIN ISO 9001



ZERTIFIKAT

Die

DQS Deutsche Gesellschaft zur Zertifizierung von Qualitätssicherungssystemen mbH

bescheinigt hiermit, daß das Unternehmen

IBM Deutschland
Bildungsgesellschaft mbH
Am Fichtenberg 1

D - 7033 Herrenberg

ein

Qualitätssicherungssystem

eingeführt hat und anwendet.

Durch ein Qualitätsaudit der DQS wurde der Nachweis erbracht,
daß dieses Qualitätssicherungssystem die Forderungen der folgenden Norm erfüllt:

DIN ISO 9001

Qualitätssicherungssysteme
Modell zur Darlegung der Qualitätssicherung
in Design/Entwicklung, Produktion, Montage und Kundendienst
(Identisch mit ISO 9001 : 1987, EN 29 001 : 1987, BS 5750 Part 1 : 1987)

Dieses Zertifikat ist gültig bis 10. Dezember 1995

Zertifikat-Registrier-Nr.: 1120 - 02

Frankfurt am Main, 11. Dezember 1992

DQS Deutsche Gesellschaft zur Zertifizierung
von Qualitätssicherungssystemen mbH · Berlin

PRÄSIDENT GESCHÄFTSFÜHRER

Geschäftsstellen: D-6000 Frankfurt 60, August-Schanz-Straße 21 A D-1000 Berlin 30, Burggrafenstraße 6

Die DQS ist Mitglied von E-Q-Net.
Die DQS ist akkreditiert (DAR - Registriernr.: TGA-ZQ-91003)

Qualität ist unsere Verpflichtung

Allgemeines

Zielgruppe

Dieser Lehrgang wendet sich insbesondere an Benutzerdienste und Unterstützungsfunktionen.

Gleichzeitig ist dieser Lehrgang Voraussetzung für alle OS/2-Programmierer-Lehrgänge.

Ziele des Lehrganges

Nach erfolgreichem Abschluß des Lehrganges sollten Sie:

- OS/2 Version 2 und seine Komponenten in ihrem Zusammenhang verstehen.
- Die Hardwarevoraussetzungen kennen und verstehen.
- Die Architektur des OS/2 Version 2 in den Grundzügen kennen.
- OS/2 Version 2 installieren und optimieren können.
- Die vielfältigen Einstellungsmöglichkeiten bei **Virtuellen DOS Maschinen** vornehmen können.
- Die drei Arten des virtuellen Booten von DOS-Versionen beherrschen.
- Die Datenorganisation auf Disketten und Festplatten kennen und die Dateien pflegen und verwalten können.
- Druckerobjekt anlegen und Drucker-Treiber installieren und richtig zuordnen können.
- Die wesentlichen Dienstprogramme von OS/2 benutzen können.
- Die grafischen Möglichkeiten und Funktionen kennen.
- Einen Überblick über die Besonderheiten der Anwendungsentwicklung unter und für Präsentations-Manager haben.

Voraussetzungen

Vor Besuch dieses Lehrganges sollten die Teilnehmer folgende Lehrgänge besucht haben oder ein gleichwertiges Wissen mitbringen:

- 78JEU, OS/2 Version 2 für Ein- und DOS-Umsteiger
- 78J8E, Praktisches Arbeiten mit OS/2 Version 2

Weitere Lehrgänge

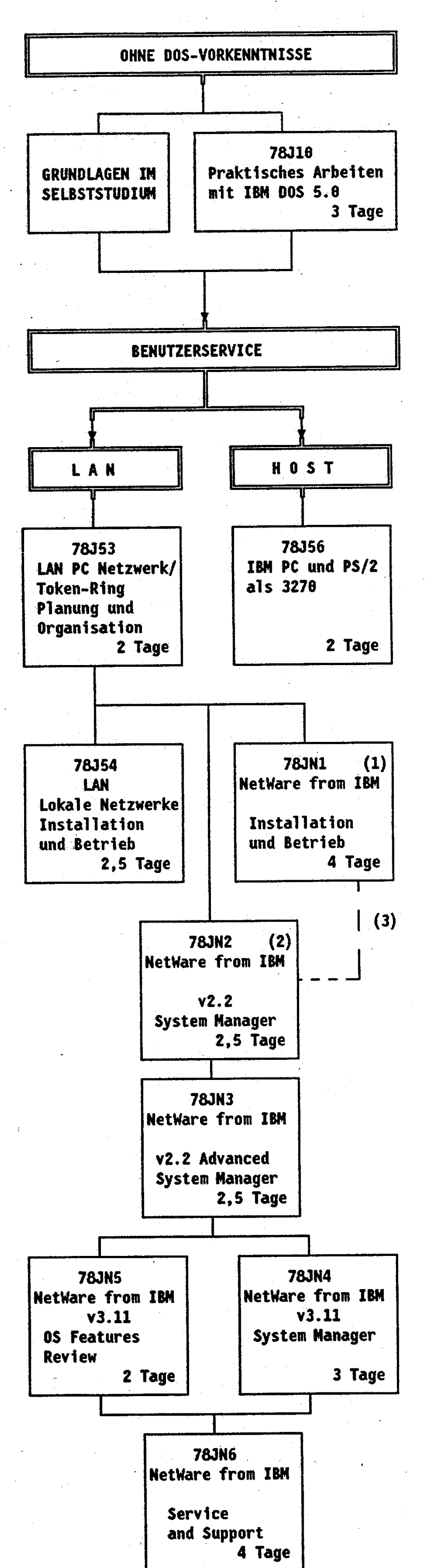
★ 17T65, SAA-CUA Konzept und Realisierung

Markierungen im Text

Wörter mit besonderer Bedeutung wurden in diesem Lehrgang wie folgt hervorgehoben:

- **Fett** sind Wörter besonderer Wichtigkeit in dem jeweiligen Zusammenhang.
- *Kursiv* sind die englischen Begriffe gekennzeichnet.
- Unterstrichen sind Auswahlen aus Menüs oder die Benutzung von Funktionstasten wie z.B. Druck.
- GROSSE Buchstaben kennzeichnen Namen von Programmen oder Dateien, welche Sie unter OS/2 benutzen können.

DOS-Curriculum

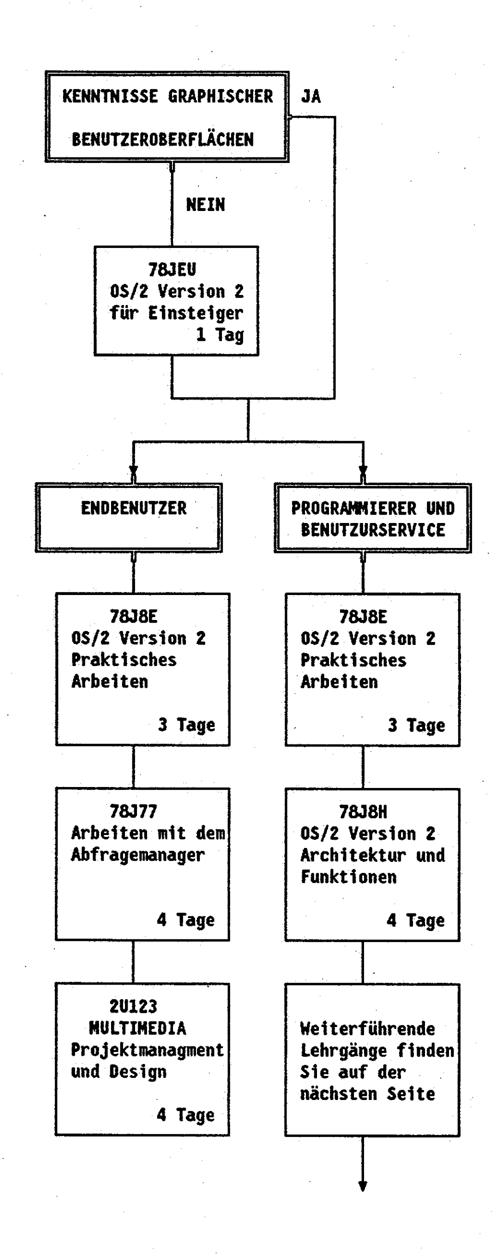


(1) Ausbildung ohne Notwendigkeit zur Prüfung als NOVELL-CNE (CNE = Certified NetWare Engineer).

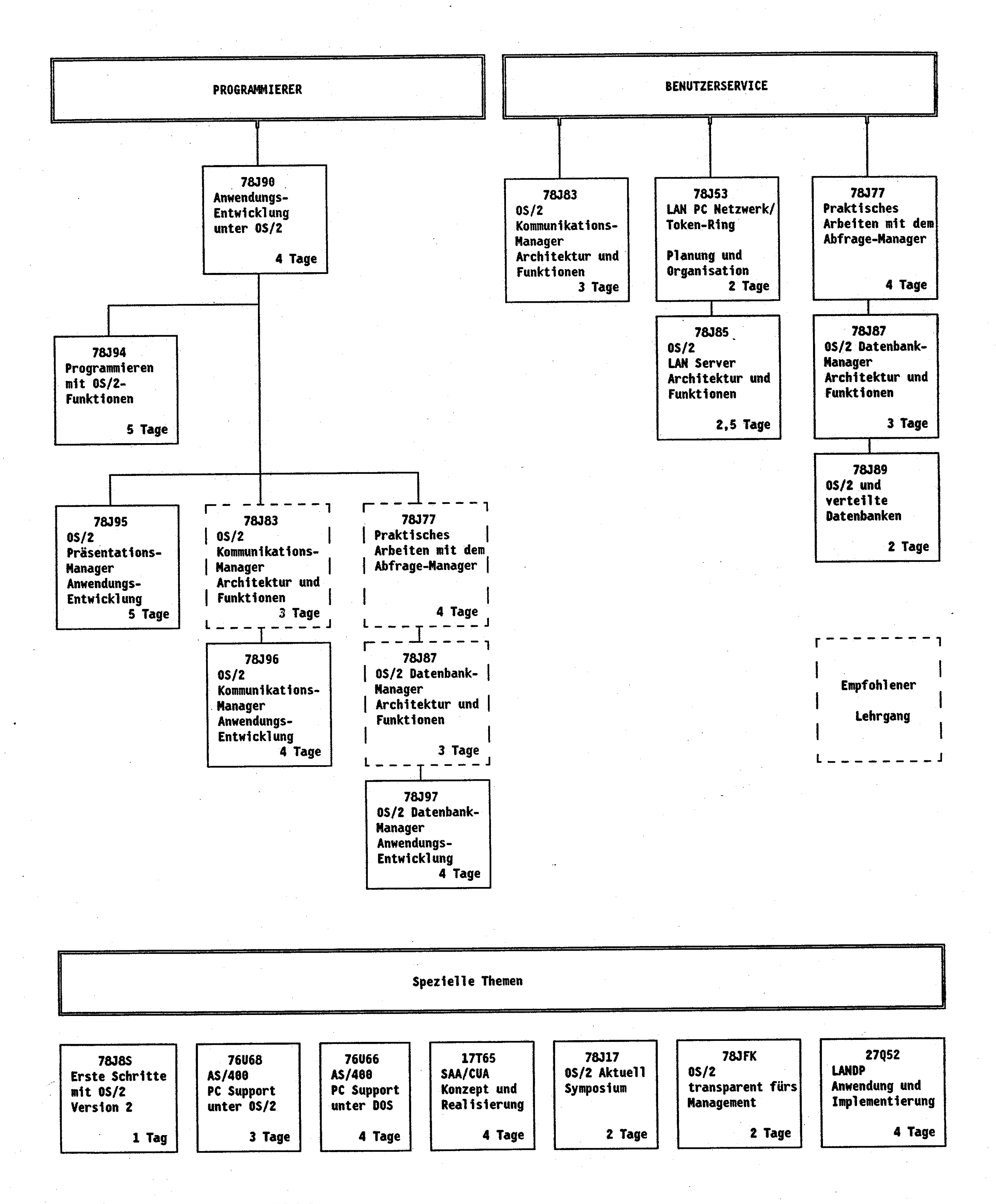
(2) Ausbildung mit/ohne Notwendigkeit zur Prüfung als NOVELL-CNE (CNE = Certified NetWare Engineer).

(3) Die Ausbildung kann weiter vervollständigt werden.

Curriculum OS/2 Basisausbildung



Curriculum OS/2 Erweiterte Ausbildung



Stand: 20. Januar 1993

Literatur

- OS/2 Version 2.0 Volume 1: Control Program, GG24-3730
- OS/2 Version 2.0 Volume 2: DOS and Windows Environment, GG24-3731
- OS/2 Version 2.0 Volume 3: Presentation Manager and Workplace Shell, GG24-3732
- OS/2 Version 2 Volume 4: Writing Programs, GG24-3774
- OS/2 Version 2.0 Volume 5: Print Subsystem, GG24-3775
- OS/2 Version 2.X : Remote Installation, GG24-3780
- OS/2 Version 2.X : Automated Installation for CID, GG24-3783
- The Design of OS/2
  H. M. Deitel, M. S. Kogan
  Addison-Wesley ISBN 0-201-54899-5
- OS/2 2.1 Integrationsplattform
  D. Hecker, H.J. Götz
  FRANZIS', ISBN 3-7723-4981-1
- OS/2 Technical Library, Application Design Guide
  Form-Nummer 10G6260
- OS/2 Technical Library, Programming Guide
  Form-Nummer 10G6261
- OS/2 Technical Library, Control Program Programming Reference
  Form-Nummer 10G6263
- CUA Advanced Interface Design Guide (SC26-4582-0, October, 1991).
- Programming the OS/2 Presentation Manager,
  Charles Petzold,Microsoft Press, ISBN 1-55615-170-5
- OS/2 Developer, G362-0001-nn
- OS/2 Monthly, JDS Publishing
- Inside OS/2
  Aktuelles Wissen Verlagsgesellschaft

Warenzeichen und Markenzeichen

- IBM ist eingetragenes Markenzeichen der IBM Corp.
- MS ist eingetragenes Markenzeichen der Microsoft Corp.
- C/2 is a trademark of International Business Machines Corp.
- dBASE II is a trademark of Ashton-Tate Corporation.
- dBASE III is a trademark of Ashton-Tate Corporation.
- ETHERAND is a trademark of International Business Machines Corp.
- Lotus 1-2-3 is a trademark of Lotus Development Corporation.
- Microsoft is a trademark of the Microsoft Corporation.
- Operating System/2 is a trademark of IBM.
- OS/2 is a trademark of IBM.
- PC Mouse is a trademark of Metagraphics/Mouse Systems.
- Personal Computer AT and AT are registered trademarks of IBM.
- Personal System/2 and PS/2 are registered trademarks of IBM.
- Presentation Manager is a trademark of IBM.
- Proprinter is a trademark of IBM.
- Quietwriter is a registered trademark of IBM.
- SAA is a trademark of IBM.
- Symphony is a trademark of Lotus Development Corporation.
- Systems Application Architecture is a trademark of IBM.
- VisiOn is a trademark of VisiCorp.
- DR DOS is a trademark of Digital Research

Inhaltsverzeichnis

Kapitel 1. OS/2 Version 2 1-1
1.1 Allgemeines 1-3
1.2 Präsentations Manager 1-7
1.3 Arbeitsoberfläche 1-10
1.4 Geänderte Funktionen 1-13

Kapitel 2. Installation OS/2 Version 2 2-1
2.1 Allgemeines 2-3
2.2 Behandlung des Boot-Managers 2-10
2.3 Installation anderer Betriebssysteme 2-19

Kapitel 3. OS/2 Version 2, Architektur HW 3-1
3.1 Architektur Hardware 3-3
3.2 Architektur Mikrokanal und Implementation 3-19
3.3 Mikrokanal Architektur 3-20
3.4 BUS Architektur des Personal Computer 3-26
3.5 System Konfiguration 3-28
3.6 Hardware als Einflußfaktor 3-31

Kapitel 4. OS/2 Version 2, Architektur OS/2 4-1
4.1 Architektur Betriebssysteme 4-3
4.2 Startvorgang von Hardware und Betriebssystem . . . 4-23

Kapitel 5. Drucken unter OS/2 Version 2 5-1
5.1 Allgemeines 5-3
5.2 Das Konzept der Druckausgabe 5-4
5.3 Verwenden von Trennseiten, separator pages 5-17
5.4 Wege des Druckjobs 5-21
5.5 Drucken im LAN 5-25
5.6 Drucken ohne Direktmanipulation 5-27

Kapitel 6. DOS und Windows unter OS/2 V2 6-1
6.1 Allgemeines 6-3
6.2 Migration von Programmen nach OS/2 PM 6-4
6.3 Virtuelle DOS-Maschinen 6-5
6.4 Speichermanagement unter DOS und WINDOWS 6-9
6.5 Booten von virtuellen DOS-Maschinen 6-21

6.6 Windows-Programme unter OS/2 Version 2 6-27
6.7 Installation von Windows-Programmen 6-31

Kapitel 7. Systemanpassung 7-1
7.1 Allgemeines 7-3
7.2 Systemoptimierung 7-7
7.3 Die Datei CONFIG.SYS 7-10
7.4 Diagnose 7-16
7.5 Dienstprogramme zum Erstellen von Speicherauszügen 7-20
7.6 Die Dienstprogramme TRACE und TRACEFMT 7-23
7.7 Das Dienstprogramm PSTAT 7-27
7.8 Das Dienstprogramm PATCH und die Korrekturdienste 7-28
7.9 Anzeige des Änderungsstandes des OS/2 7-29
7.10 Tips und Hinweise zu OS/2 Version 2 7-30

Kapitel 8. Datenbestandsverwaltung 8-1
8.1 Allgemeines 8-3
8.2 Plattenorganisation 8-5
8.3 Das High Performance File System 8-7
8.4 Anlegen und Löschen von Dateien 8-14
8.5 Sichern von Datenbeständen 8-22

Kapitel 9. Startmöglichkeiten 9-1
9.1 Allgemeines zu Stapeldateien 9-3
9.2 Systemstart (STARTUP) 9-8

Kapitel 10. Grafikfunktionen des PM 10-1
10.1 Schriftarten unter OS/2 10-3
10.2 Rastergrafik und Vektorgrafik 10-6
10.3 Grafische Grundfunktionen 10-8
10.4 Darstellungs-Bereich und Koordinatensysteme . . 10-10
10.5 Grafik-Segmente und Metafiles 10-14
10.6 Einheiten-Kontext 10-15
10.7 Das Dienstprogramm "Bilddateien anzeigen" (PICVIEW) 10-16
10.8 OS/2 Grafik-Anwendung PM Chart 10-17

Kapitel 11. Überblick Anwendungsentwicklung 11-1

11.1 Erweiterter Editor EPM 11-3
11.2 Voraussetzungen Software 11-8
11.3 TOOLKIT - Programming Tools and Information . . 11-9
11.4 Informations-Entwicklung 11-13
11.5 IPF-Tag-Übersicht 11-26
11.6 Programmierung für OS/2 11-29
11.7 Unterstützende Programme 11-35
11.8 Die Prozedurensprache REXX 11-36

Anhang A. Anhang spezielle Übungen A-1
A.1 Dienstprogramme A-11
A.2 Der Symboleditor A-13
A.3 Weitere Dienstprogramme A-14

Anhang B. Abkürzungen aus dem US-Englischen B-1

Anhang C. Index C-1

Abbildungsverzeichnis

1-1. Jedes Ding hat seine drei Seiten... (J8HG0102) 1-3
1-2. Geschichte des OS/2 (JFKG0201) 1-4
1-3. Übersicht OS/2-Komponenten (J8HG9917) 1-5
1-4. Architektur-Modell OS/2 und Präsentations Manager(J8HG0110) 1-9
1-5. Objekt - Aktion (J8HG0100) 1-12
2-1. Vorplanung: Welche Betriebssysteme? (J8HG0205) 2-6
2-2. Bereitstellung der gewünschten Betriebssysteme (J8HG0210) 2-7
2-3. Mögliche Anordnung des Boot-Manager (J8HG0215) 2-9
2-4. Mögliche Anordnung der Plattenbereiche (J8HG0220) 2-15
2-5. Verwendung Bezeichnung Logischer Laufwerke 1 (J8HG0225) 2-16
2-6. Verwendung Bezeichnung Logischer Laufwerke 2 (J8HG0230) 2-17
2-7. Verwendung Bezeichnung Logischer Laufwerke 3 (J8HG0235) 2-18
2-8. SEIMAGE Verzeichnis Struktur (J8HG0236) 2-24
3-1. Instruktions-Hierarchie Intel-Prozessoren (J8HG9901) 3-5
3-2. AT-Bus und Mikrokanal (J8HG2107) 3-21
3-3. Der Mikrokanal 3-24
3-4. Der PC BUS 3-26
4-1. Beziehungen Software-Hardware unter DOS (J8HG9910) 4-4
4-2. Beziehungen Hardware-Software unter OS/2 (J8HG9911) 4-5
4-3. Schutzring-Konzept (J8HG9913) 4-7
4-4. Adress-Umsetzung (J8HG9902) 4-10
4-5. Shared Memory (J8HG9914) 4-11
4-6. Funktion von Queues (J8HG9918) 4-12
4-7. Named Pipes, 3 Kanäle und Duplex (J8HG9919) 4-13
4-8. Paging-Konzept, Virtueller Speicher (J8HG9903) 4-16
4-9. Thunk layer (J8HG9915) 4-22
4-10. Speicher-Rekonfiguration (J8HG0350) 4-23
5-1. Viele Anwendungen - viele Druckerobjekte - mehrere Drucker (J81G0411) 5-5
5-2. Kontext-Menü eines Druckerobjektes (J81G0401) 5-6

5-3. Festlegung des zu verwendenden Drucker-Treibers (J8HG0402) 5-7
5-4. Kontext-Menü eines Drucker-Treibers (J8HG0403) . . 5-8
5-5. Notizbuch zum Kopieren eines Druckerobjektes (J8HG0405) 5-11
5-6. Auswahl Erstellen Drucker (J8HG0406) 5-11
5-7. Ausschnitt aus dem Ordner Schablonen(J8HG0407) 5-12
5-8. Wege eines Druckjobs (J8HG2104) 5-22
5-9. LAN - Ordner (J8HG0408) 5-25
5-10. Netzlaufwerke (J8HG0409) 5-25
5-11. Netzdrucker (J8HG0410) 5-26
6-1. Migration von DOS nach OS/2 (J8HG1107) 6-4
6-2. Konzept der Virtuellen DOS-Maschinen (J8HG0505) . 6-6
6-3. DOS-Programm an der Arbeitsoberfläche (J8HG0506) 6-8
6-4. Lage der UMB's (J8HG0511) 6-10
6-5. Lage der HMA (J8HG0513) 6-11
6-6. Funktion des EMS (J8HG0514) 6-12
6-7. Dialogfenster für die DOS-Einstellungen (J8HG0512) 6-14
6-8. Speicherverwaltung in einer VDM (J8HG0510) 6-16
6-9. Konzept der Zwischenablage (J8HG0508) 6-29
6-10. Konzept des DDE (J8HG0540) 6-30
7-1. OS2.INI und Referenz (J8HG2106) 7-4
7-2. Geltungsbereich der CONFIG.SYS (J8HG0602) 7-10
8-1. Beispiel eines Ordners "Dateiverwaltung" (J8HG0701) 8-4
8-2. Konzept des HPFS (J81GHPFS) 8-9
8-3. Arbeitsoberfläche-Struktur ist eingebunden in Plattenstruktur (J8HG0303) 8-15
8-4. Syntaxübersicht UNDELETE (J8HTUDEL) 8-17
8-5. Anzeige wiederherstellbarer Dateien 8-18
8-6. Syntaxübersicht BACKUP (J8HTBACK) 8-22
8-7. Syntaxübersicht RESTORE (J8HTREST) 8-24
9-1. Syntaxdiagramm Befehl START (J8HTSTRT) 9-9
9-2. Stapeldatei, Beispiel 1 9-10
9-3. Der Ordner Systemstart (J8HG0810) 9-12
10-1. Prinzip des ATM (J8HG0612) 10-4
10-2. Der Weg eines Zeichens zum Bildschirm (J8HG1001) 10-6
10-3. Grafische Grundfunktionen (J8HG1002) 10-8
10-4. Hardware-unabhängiges Koordinatensystem (J8HG1003) 10-10

10-5. Hardware-bezogenes Koordinatensystem (J8HG1004) 10-11
10-6. Hardware-bezogenes Koordinatensystem (J8HG1005) 10-12
10-7. Presentation Space und Page Viewport (J8HG1006) 10-13
10-8. Grafikelemente (J8HT1001) 10-14
10-9. Zuordnung Anzeigebereich zu Ausgabe-Einheit (J8HG1008) 10-15
10-10. Dienstprogramm PICVIEW (J8HG1010) 10-16
10-11. Dienstprogramm PM Chart mit Liniengrafik (J8HG1020) 10-18
10-12. PM Chart mit Tabelle und Grafik (J8HG1022) 10-22
10-13. PM Chart mit Zeichnung (J8HG1024) 10-24
11-1. Erweiterter Editor (J8HG0905) 11-3
11-2. Font Editor (J8HG1101) 11-10
11-3. Konzept Erstellung Schriftarten-Datei (J8HG1102) . 11-11
11-4. Dialog Editor (J8HG1103) 11-12
11-5. Konzept des IPF (J8HG1104) 11-13
11-6. Kompilierung einer Informations-Datei 11-25
11-7. Konzept der Programme unter PM (J8HG1105) . . . 11-30
11-8. SOM umfaßt den Kern und den PM (J8HG1108) . . . 11-32
11-9. Hierarchie der Workplace Shell Classes 11-34
A-1. Hauptfenster Symboleditor (J8HG0910) A-13

Kapitel 1. OS/2 Version 2

Ziele dieses Kapitels

Dieses Kapitel macht Sie mit OS/2 Version 2.1 bekannt.

Referenzen

- Red Book IBM OS/2 Version 2.0, Volume 3: Presentation Manager, GG24-3732
- Red Book IBM OS/2 Version 2.0, Volume 1: Control Program, GG24-3730

Inhalt dieses Kapitels

1.1 Allgemeines 1-3
1.2 Präsentations Manager 1-7
1.3 Arbeitsoberfläche 1-10
1.4 Geänderte Funktionen 1-13

1.1 Allgemeines

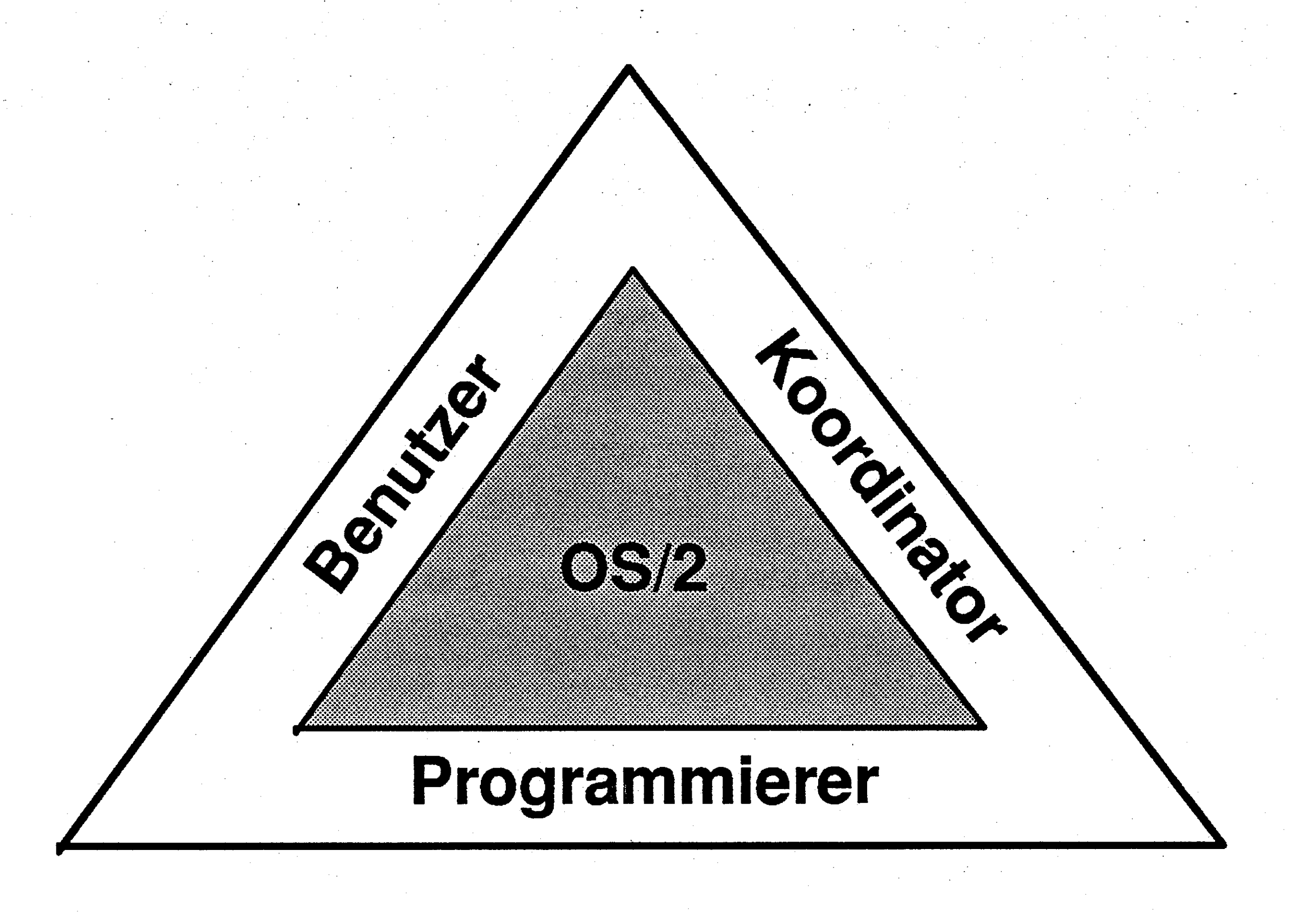


Abbildung 1-1. Jedes Ding hat seine drei Seiten... (J8HG0102)

Die Versionen 2 des OS/2 stellen bedeutende Weiterentwicklungen der Versionen 1 dar. Neben vielen Betriebssystem-internen Funktionen (wie beispielsweise 32-bit-Speicherverwaltung) wurde insbesondere die Benutzeroberfläche, die Arbeitsoberfläche, neu gestaltet.

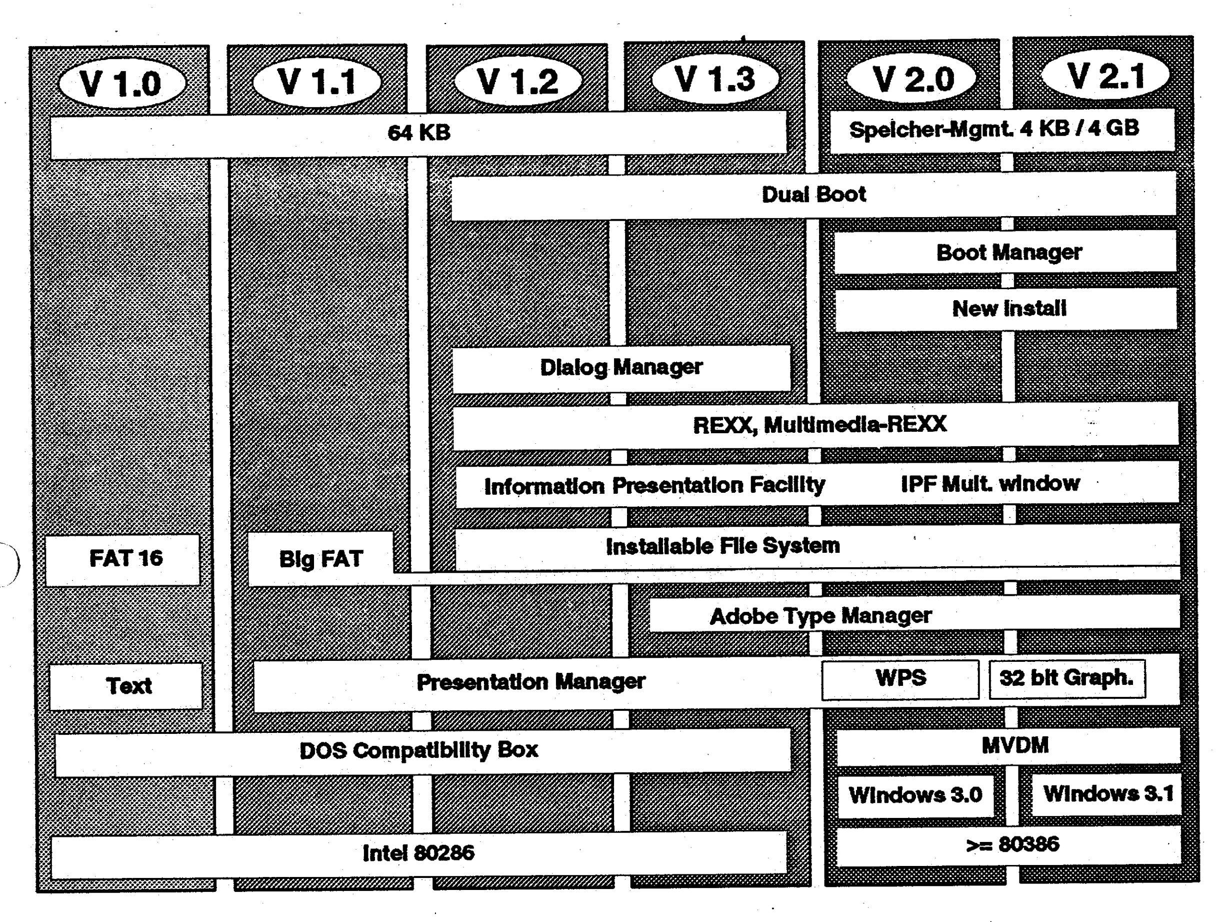


Abbildung 1-2. Geschichte des OS/2 (JFKG0201)

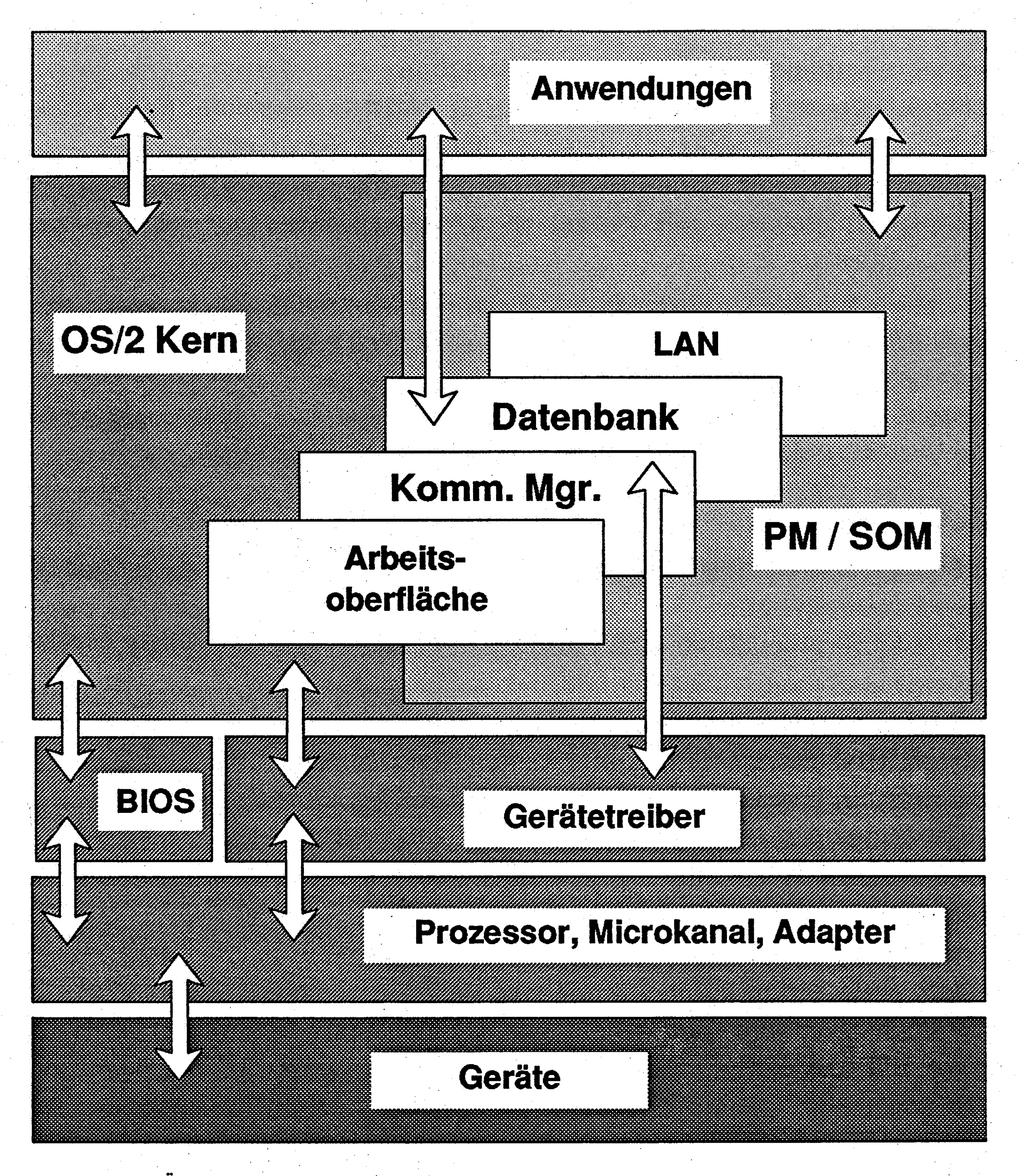


Abbildung 1-3. Übersicht OS/2-Komponenten (J8HG9917)

Es wurde Wert darauf gelegt, die bisherigen OS/2-Programme weiterverwenden zu können. Wesentliche Unterschiede sind:

- Basisversion vergleichbar mit Standard-Ausgabe,
- Zusätzliche Dienste wie Datenbank- (DB2/2) und Kommunikations-Manager (CM/2) als separate Programme lieferbar (Extended Services),
- Volle Unterstützung der 32-bit-Prozessoren ab 80386,
- Neue Benutzeroberfläche mit verstärkter Objekt-Orientierung entsprechend SAA-CUA 91,
- Hohe Flexibilität durch funktions-/anwendungsbezogene Objekte auf der Arbeitsoberfläche
- Keine festgeschriebene Hierarchie
- Individuelle Anpassung der Druckausgaben
- Flexible Verwaltung von Dateiobjekten über das Objekt „Laufwerke“
- Unterstützung von DOS-Programmen durch bis zu 240 Virtuelle DOS-Maschinen (MVDM)
- Unterstützung von Windows-Programmen durch OS/2-Windows-Funktionen,
- Erweiterter Lieferumfang der Basisversion mit:
  - Leistungsstarkem Erweiterten Editor,
  - Produktivitätshilfen wie Uhr, Kalender, Durchsuchen mehrerer Diskettenlaufwerke und Festplatten u.a.m.
  - Unterstützung mehrerer Betriebssysteme auf einer Festplatte durch den *Boot Manager*,
  - Neu gestaltete Installations-Prozedur mit vielfältigen Möglichkeiten der Organisation und Optimierung,
  - Neu gestaltetem Einführungsprogramm,
  - Umfangreiche Hilfen durch On-line-Handbücher,
  - Unterhaltungs-Programme zum spielerischen Einüben wichtiger Benutzungsfunktionen,

1.2 Präsentations Manager

Den Präsentations Manager gibt es seit der OS/2 Version 1.1. Vor seiner Existenz war OS/2, wie DOS, befehlszeilenorientiert. Er stellte erstmalig eine grafische Benutzeroberfläche zur Verfügung und ist seitdem das Element, das Oberflächenaktivitäten nach unten in die Anwendungen, Sitzungen und in das Betriebssystem transportiert und natürlich auch in umgekehrter Richtung.

Mit Herausgabe der OS/2 Version 2.0 wurde der Oberflächenteil wieder aus dem Präsentations Manager herausgenommen und der Arbeitsoberfläche zugeordnet.

Somit hat der Präsentations Manager eine andere Bedeutung erhalten.

- Er steuert Eingabe-/Ausgabe-Operationen
- Erlaubt/steuert die *drag and drop*-Funktion
- Überwacht alle auf der Arbeitsoberfläche befindlichen Objekte
- Er stellt, abgesehen vom Betriebssystem die oberste Hierarchiestufe dar.
- Er ist das GUI von OS/2
- Erlaubt Programmen ein entsprechendes *User Interface*, da die Steuerungselemente Teile des PM sind und nicht vom Programm gestellt werden.
- Er stellt das strategische Element entsprechend der IBM SAA dar und entspricht bis ins Detail den Spezifikationen des aktuellen CUA-Standards
- Grundsätzlich gibt es 2 verschiedene API's:
  - Fenster-API, das seinen Ursprung in der Fenstertechnik von Microsoft-WINDOWS hat.
  - Grafisches API, das die Techniken von
    - IBM Graphics Data Display Manager (GDDM)
    - IBM 3270 Graphics Control Program (GCP)
    - Microsoft Windows Graphics Device Interface (GDI)

    integriert.

Funktionell ist der Präsentations Manager eine Erweiterung der OS/2 Architektur und läuft oberhalb des OS/2-Kernels in der PM-Sitzung. Er besteht aus

einer Anzahl DLL's und ausführbaren Programmen. Der Steuerungsdatenstrom läuft in Form von *messages* ab. Es gibt eine Vielzahl verschiedener, z.B.:

- WindowMessages
- MouseMoveMessages
- ScrollBarMessages

Der Präsentations Manager hält einen eigenen Bildschirm Pufferbereich bereit, dieser enthält nur die Daten, die auf dem Bildschirm zu sehen sind.

Für den Wiederaufbau eines verdeckt gewesenen Bereiches ist nicht der PM verantwortlich, er gibt lediglich mithilfe des Fokus die Information an die Sitzung, daß diese ihr Bild wieder aufbauen muß. Dies spart Speicherplatz. Das Modell des in die Architektur eingebundenen Präsentations Managers wird dargestellt in Abb. 1-4 auf Seite 1-9

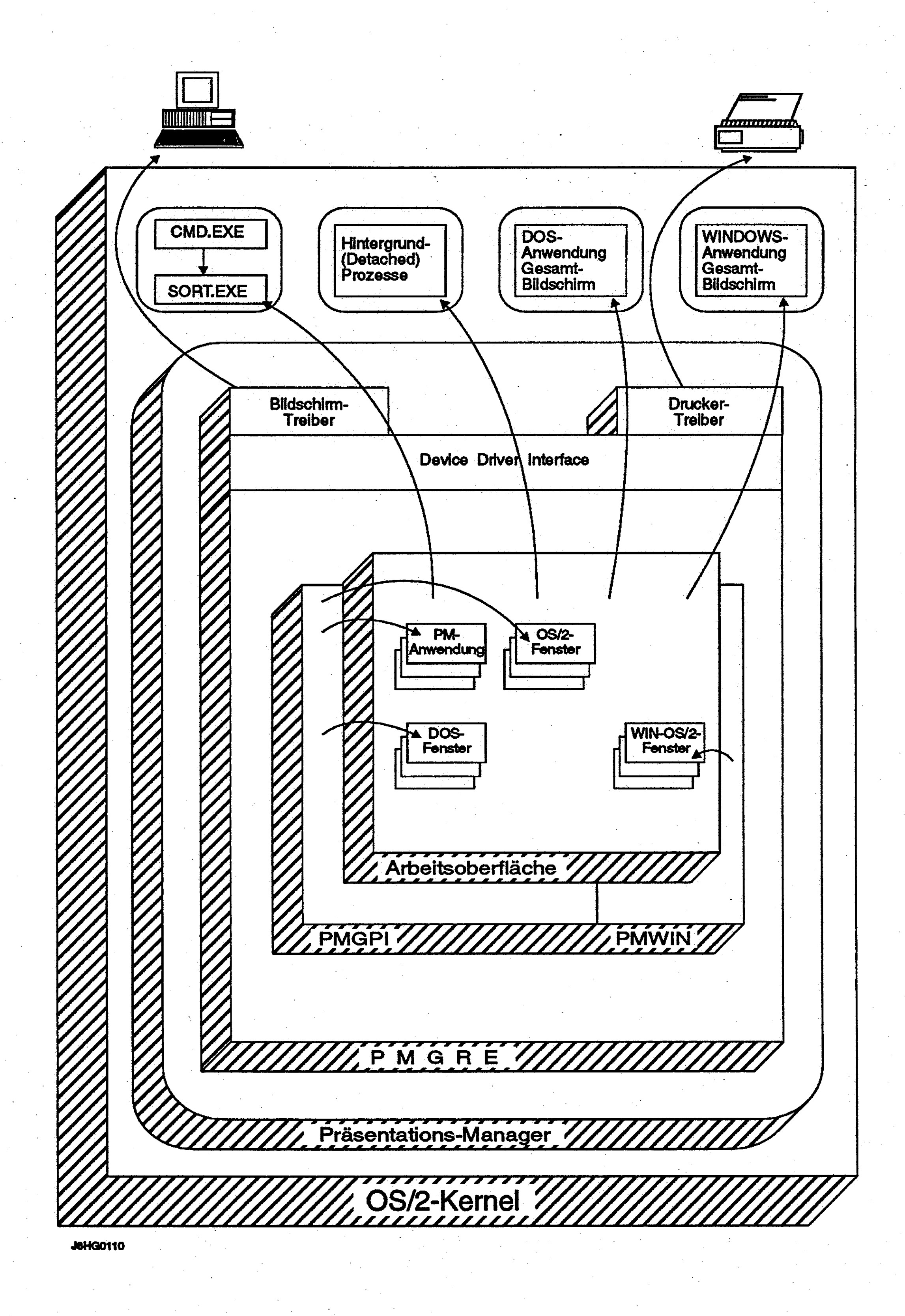


Abbildung 1-4. Architektur-Modell OS/2 und Präsentations Manager(J8HG0110) ,

1.3 Arbeitsoberfläche

Die bisherige Benutzeroberfläche des OS/2 (grafisch, Fenster-orientiert) wurde durch eine komplette Neuentwicklung ersetzt. Wir haben jetzt eine:
Objekt-orientierte grafische Benutzeroberfläche
Diese wird auch *OOGUI - Object-oriented Graphical User Interface* genannt.

Diese Neuentwicklung entspricht dem fortgeschriebenen Standard von SAA für die Benutzeroberfläche, dem CUA 1991. Es wird das sogenannte *Workplace Model* verwirklicht, welches dem Benutzer die **Arbeitsoberfläche** *(Workplace Shell)* darstellt.

Die Änderungen sind so gravierend, daß die Terminologie weitgehend neu erlernt werden muß. Vergleiche mit OS/2 Versionen 1.x sind nicht hilfreich.

Engl. Begriffe	Deutsche Begriffe und Beschreibung
Container	**Behälter**, Allgemeinbegriff für ein Objekt, das Programm- und Dateieinträge beinhalten kann.
Folder	**Ordner**, ein Objekt, das selbst weitere Einträge enthalten kann.
Work area	**Arbeitsordner**, ein Objekt, das selbst weitere Einträge enthalten kann. Im Gegensatz zum einfachen Ordner spezielles Verhalten.
Object	**Objekt**, eine auf der Oberfläche individuell ansprechbare Einheit.
Program Object	**Programmobjekt**, repräsentiert ausführbare Datei.
Data Object	**Datenobjekt**, enthält Daten in beliebiger Form.
Device Object	Geräteobjekt, ein Objekt, das Zugriff zu einer physischen Einheit ermöglicht
Printer Object	**Druckerobjekt**, das Druckersymbol. Es stellt einen logischen Drucker dar.
Shredder	**Papierkorb**, Objekt, das zum Löschen von anderen Objekten dient.
Notebook	**Notizbuch**, dient zum Einstellen von Parametern. Der Umfang ist Objekt-abhängig.

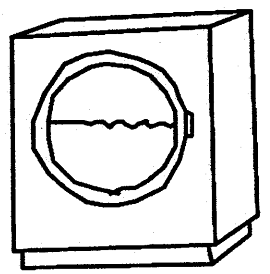
Value Set Angebot mehrerer Auswahlmöglichkeiten, z.B.: Werkzeuge in einem Zeichenprogramm.

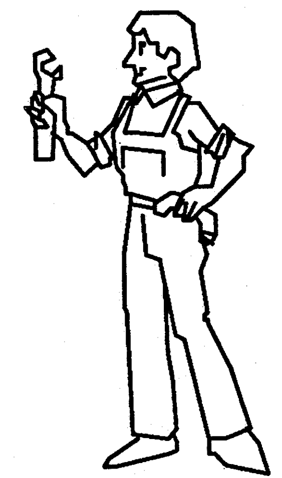
Auffallend bei der Benutzung ist das fast völlige Fehlen vorgegebener hierarchischer Strukturen auf der **Arbeitsoberfläche** *(desktop)*. Dinge, welche bisher unter dem „Dach" eines Programmes zusammengefaßt wurden, sind nun selbständige Objekte; z.B. gibt es jetzt mehrere Druckerobjekte anstelle eines Druck-Managers mit mehreren Warteschlangen.

Wichtig!

Alle Arbeit mit OS/2 folgt dem Prinzip:







Aktion

Abbildung 1-5. Objekt - Aktion (J8HG0100)

Was auch immer Sie machen:

Sie brauchen erst ein Objekt und dann können Sie damit machen, was dieses Objekt zuläßt!

1.4 Geänderte Funktionen

Viele der Funktionen von OS/2 1.3 sind durch anders geartete Funktionen ab OS/2 2.0 ersetzt worden. Folgende Funktionen der Benutzeroberfläche sind besonders hervorzuheben:

1. System-Manager
2. Druck-Manager
3. Datei-Manager

Die geänderten Funktionen und ihr Ersatz werden im folgenden kurz besprochen.

System-Manager

Die bisherige hierarchische Anordnung des System-Manager wurde durch eine Anzahl **Ordner** *(folder)* ersetzt, welche beim Systemstart auf der **Arbeitsoberfläche** *(desktop)* liegen. Dadurch ist eine größere Flexibilität gegeben.

Druck-Manager

Das Konzept des Druckens wurde weiter an die Erfahrungen aus der Welt der Großsysteme angeglichen. Gleichzeitig wurden Vereinfachungen vorgenommen. Das Ziel einer jeden Druckausgabe wird jetzt als rein logischer Drucker, als **Druckerobjekt**, festgelegt. Diese logischen Drucker lassen sich als Druckerobjekte jederzeit leicht und logisch an geänderte Gegebenheiten, wie einen anderen Drucker oder eine andere Schnittstellen-Zuordnung, anpassen. Vieles davon läßt sich mit der **Direktmanipulation** *(drag and drop)* durchführen.

Datei-Manager

Die Verwaltung von Festplatten und Disketten wird jetzt über das Objekt **Laufwerke**, *(Drives)* durchgeführt, welches die logischen Laufwerke repräsentiert. Dieses Objekt ist selbst ein Ordner, welcher weitere Ordner enthält. Dadurch ist die hierarchische und zentrale Steuerung zugunsten flexiblerer Anordnungen und Arbeitsweisen entfallen.

Näheres finden Sie in Kapitel 8, „Datenbestandsverwaltung“ auf Seite 8-1.

Kapitel 2. Installation OS/2 Version 2

Ziele dieses Kapitels

In diesem Kapitel lernen Sie:

- Wie Sie eine Installation des OS/2 Version 2 planen und vorbereiten.
- Wie OS/2 Version 2 installiert wird.

Inhalt dieses Kapitels

2.1 Allgemeines 2-3
2.1.1 Aufteilung der Festplatte(n) 2-4
2.2 Behandlung des Boot-Managers 2-10
2.2.1 ÜBUNG: Nachinstallation des Bootmanagers . . . 2-11
2.2.2 ÜBUNG: Arbeit mit dem Dienstprogramm "setboot" 2-13
2.2.3 ÜBUNG: Erweiterter Modus des Bootmanagers . 2-14
2.3 Installation anderer Betriebssysteme 2-19

2.1 Allgemeines

OS/2 Version 2.1 wird mit folgendem Material geliefert:

- Handbücher:
  - Achtung - Bitte zuerst lesen, Teilenummer 63G3907
  - Benutzerhandbuch, Teilenummer 63G3905
  - Installationshandbuch, Teilenummer 63G3904
  - Tastaturen, Teilenummer 63G3903
  - OS/2 Applications ... Direct to you, Teilenummer 63G3227
  - Schutzrechte und Nutzungsumfang, Teilenummer 45F0352
  - Kurzübersicht, Teilenummer 63G3906
  - Zusatzerklärung für Serviceleistungen, Teilenummer 63G3908
  - Registrierungskarte, Teilenummer 67G8537
  - Inside OS/2 Test-Angebot, Teilenummer 63G4103
  - IBM Gutschein für Produkt, Teilenummer 39G5235
  - Adress-Aufkleber für Registrierungsadressen, Teilenummer 63G3909/1/2/3
- 21 Disketten, diese belegen ca. 15 MB bis 35 MB Festplattenplatz.

Es gibt die Möglichkeit, mehrere Betriebssysteme (z.B. DOS 5, OS/2 1.3 usw.) auf eine Festplatte zu laden. Dazu wird der *Boot-Manager* verwendet. Dieser belegt einen eigenen, nicht vom Benutzer zu formatierenden Plattenbereich von 1 MB, welcher von keinem Programm aus ansprechbar ist.

Die Installation wird mit einem **Mini-OS/2**, welches von den ersten beiden Disketten geladen wird, durchgeführt. Enthalten in der Installation ist der Aufruf eines *full screen FDISK*. Damit läßt sich eine Festplatte aufteilen und der zu ladende Bereich (*partition*) festlegen.

2.1.1 Aufteilung der Festplatte(n)

Entsprechend den Erfordernissen des späteren Betriebes muß eine mehr oder weniger aufwendige Planung gemacht werden. Für den Normalfall sind viele der folgenden Überlegungen von geringer Bedeutung, während eine Installation im Bereich der Unterstützung oder Programmierung ähnlich aussehen kann.

Grundsätzliches

Eine Festplatte muß, bevor sie formatiert und damit benutzbar gemacht wird, als erstes einmal **partitioniert** werden. Was bedeutet **partitionieren**?

1. Die Festplatte wird in Datenräume aufgeteilt, deren Größe der Anwender oder Benutzerservice zu definieren hat.
2. Aus der Entwicklung des DOS heraus, ist es erforderlich, daß eine Festplatte eine **Primäre Partition** aufweist. Nur von dieser **Partition** aus konnte bisher das Betriebssystem z.B. DOS gestartet werden. Sie hat den Laufwerksbuchstaben „C:".
3. Eine Festplatte kann, wenn der Benutzer das so will, nur in eine einzige **Partition** aufgeteilt werden. Was allerdings bestimmte Nachteile mit sich bringt, auf die wir noch zu sprechen kommen.
4. Zusätzlich kann eine **Erweiterte Partition** angelegt werden, die wiederum in weitere **Logische Laufwerke** aufgeteilt werden kann.
5. Jedes logische Laufwerk erhält automatisch den nächsten im Alphabet vorkommenden Buchstaben als **Laufwerksbezeichnung**.
6. Solange Sie nur eine physische Festplatte haben, ist das ganz simpel, Weil einfach weiterge-„gezählt" wird. Also:

LW-Buchstabe	Partition, resp. logisches Laufwerk
C:	als **Primäre Partition**

D:	als **Erweiterte Partition**, oder als erstes **Logisches Laufwerk** in der **Erweiterten Partition**
E:	als zweites **Logisches Laufwerk** in der **Erweiterten Partition**

Dies Spiel können Sie natürlich noch weiter treiben, je nachdem wieviele **Logische Laufwerke** Sie einrichten.

7. Ein bißchen komplizierter wird es, wenn Sie eine zweite physische Festplatte partitionieren wollen. Jede Festplatte „erwartet" eine **Primäre Partition**. Wenn Sie nun diese partitionieren und eine solche anlegen, wird dieser automatisch der Buchstabe „D:" zugeordnet. Da jeder Buchstabe nur einmal vorkommen kann, hat dieses zur Folge, daß die Buchstabenvergabe Ihres ersten physischen Laufwerkes verändert wird.

8. Wenn Sie mit dem **OS/2-FDISK** partitionieren, können Sie definieren, daß Sie auf der zweiten **physischen Platte** nur eine **Erweiterte Partition** anlegen wollen. Dann bleibt Ihre fortlaufende Laufwerksbezeichnung erhalten.

Planung der Installation

Es läßt sich kein einfaches Ablaufdiagramm für die Planung und Installation machen. Die folgenden Bilder sollen bei dem Erstellen einer Liste oder Grafik helfen.

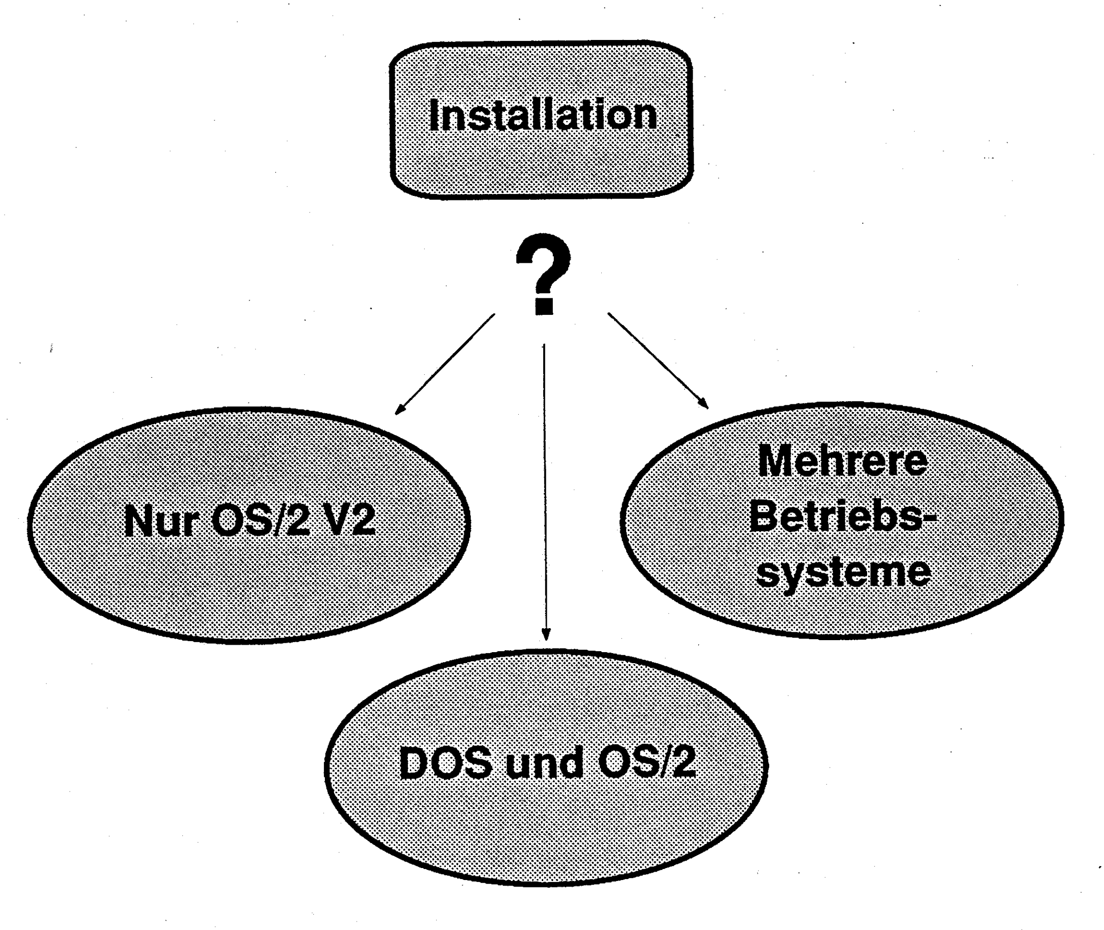


Abbildung 2-1. Vorplanung: Welche Betriebssysteme? (J8HG0205)

Vor der Installation mehrerer Betriebssysteme sollten alle Disketten und Beschreibungen vorhanden sein. Nur so kann sichergestellt werden, daß der Plattenplatz sinnvoll aufgeteilt wird.

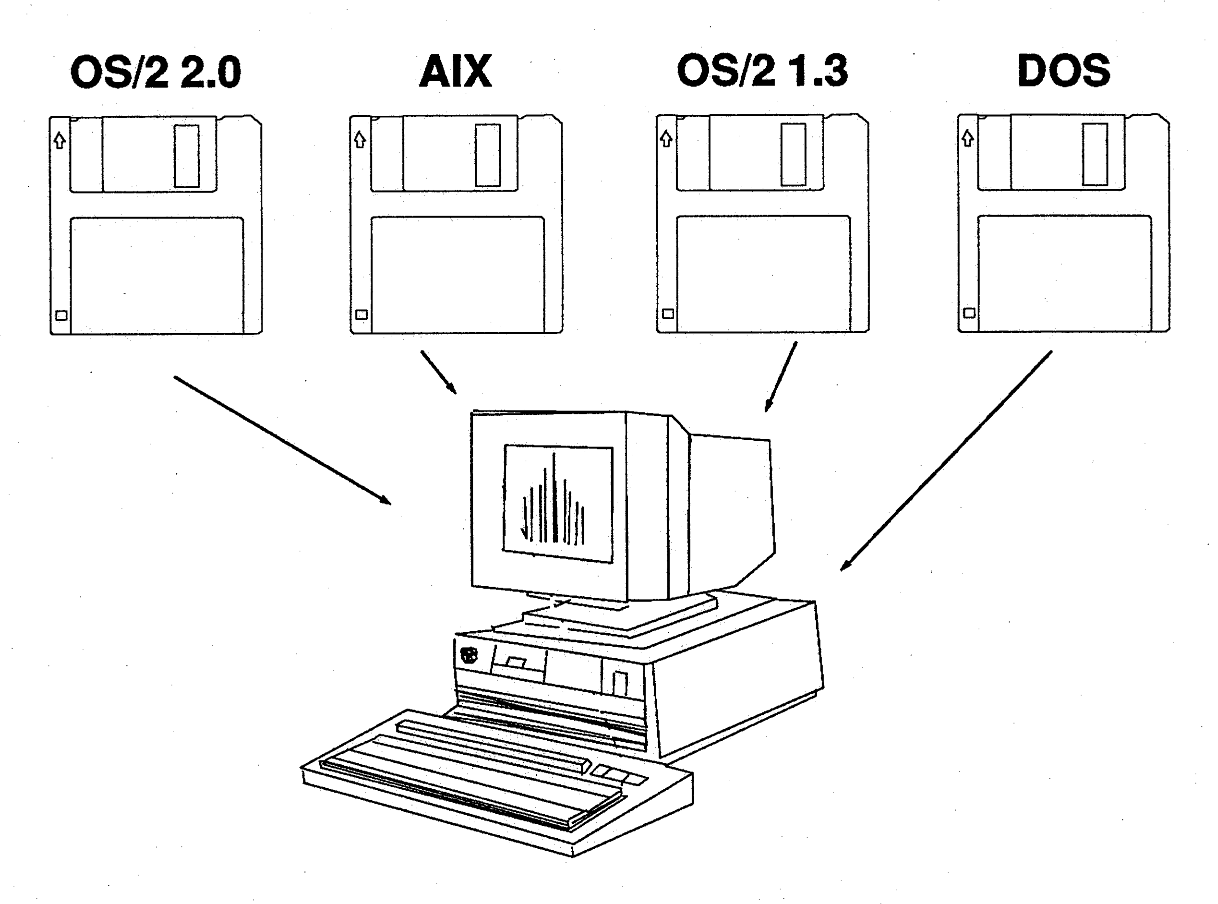


Abbildung 2-2. Bereitstellung der gewünschten Betriebssysteme (J8HG0210)

Der erforderliche Plattenplatz kann wie folgt sein:

1. OS/2 V2 - 40 MB (ohne SWAP, SPOOL und Extended Services)
2. OS/2 V1.3 - 20 MB (ohne SWAP und SPOOL)
3. DOS V5 - 5 MB
4. AIX und/oder Programme für DOS und OS/2 - 200 MB
5. CAD - Anwendungen ca. 200 - 400 MB
6. DTP - Anwendungen ca. 100 MB
7. Multimedia:
    - Entwicklungsprogramm ca. 30 MB

- Anwendung ca. 30 - 50 MB

Laden des Boot-Manager

Der Boot-Manager ist nur erforderlich, wenn mehrere Betriebssysteme geladen werden sollen. Bei einer bereits vorhandenen Installation kann der Boot-Manager an das Ende der Festplatte gelegt werden, wodurch nur der letzte Plattenbereich gelöscht und neu angelegt werden muß.

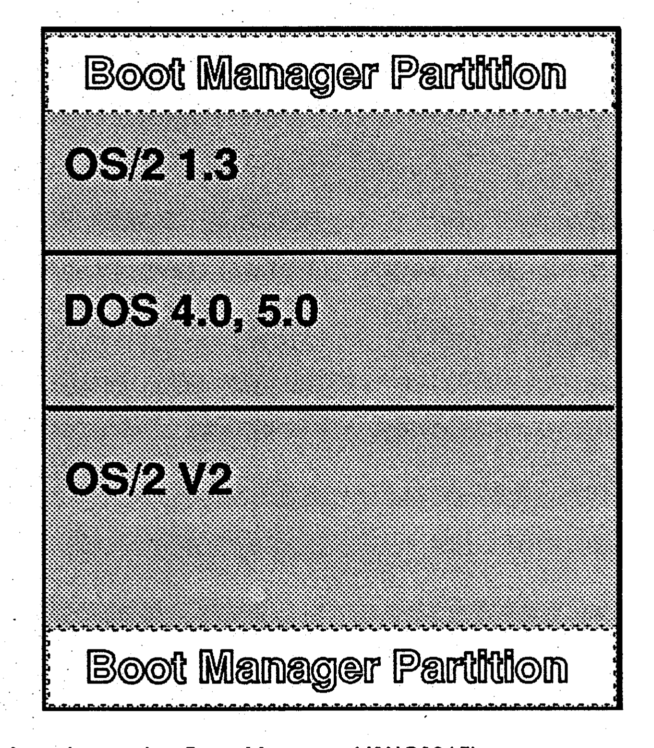


Abbildung 2-3. Mögliche Anordnung des Boot-Manager (J8HG0215)

2.2 Behandlung des Boot-Managers

Der Boot-Manager ist unabdingbare Voraussetzung für das Vorhandensein von mehreren Betriebssystemen auf einer Maschine. Da wir die Installation lediglich theoretisch behandelt haben, arbeiten wir nun heraus, wie dieser nachträglich installiert werden kann.

1. Er muß auf der 1. physischen Festplatte installiert werden. Wenn Sie versuchen, ihn auf eine 2. physische Festplatte zu installieren, werden Sie feststellen, daß diese Möglichkeit nicht gegeben ist.
2. Wo Sie ihn auf der 1. physischen Platte installieren, ist Ihnen im Rahmen der Möglichkeiten, die Ihnen das FDISK-Menü bietet, überlassen.
3. Sie können ihn daher jeweils am Ende oder auch Anfang eines freien Bereiches auf der Platte installieren.
4. Was heißt „freier Bereich"? Hiermit ist ein Bereich auf der Platte gemeint, der nicht partitioniert ist.
5. Wenn Sie einen solchen Bereich nicht haben, gibt es nur die Möglichkeit, eine bestehende Partition resp. ein logisches Laufwerk zu löschen. In diesem Fall steht Ihnen dann ein freier Bereich zur Verfügung.

2.2.1 ÜBUNG: Nachinstallation des Bootmanagers

In vielen Fällen wurde der *Boot-Manager* nicht installiert. Sie möchten ihn daher nachinstallieren.

Ziel der Übung

Sie lernen, wie man mithilfe des OS/2 FDISK die Nachinstallation durchführen kann.

Durchführung

Als erstes rufen Sie den FDISK auf.

- Öffnen Sie bitte den Ordner Befehlszeilen.
- Aktivieren Sie ein OS/2-Fenster.
- Geben Sie ein:

  `FDISK`

1. Nun wählen Sie das Laufwerk aus.
   - Gehen Sie mit dem Cursor auf das letzte logische Laufwerk der 1. physischen Festplatte und drücken Sie die ENTER-Taste.
2. Die Partition muß gelöscht werden.
   - Nun wählen Sie aus Partition löschen und drücken erneut ENTER
   - Jetzt ist der erste Menüpunkt Boot-Manager installieren aktivierbar. Bitte wählen Sie diesen aus und bestätigen Sie mit ENTER.
3. Die Frage, ob die Installation am Anfang oder Ende des freien Bereiches erfolgen soll, beantworten Sie bitte mit:

   `Am Ende des freien Bereiches`
4. Sie können nun über den Menüpunkt Startwerte festlegen das Verhalten Ihres Boot-Managers beeinflussen.

5. Um Ihre Betriebssysteme auch über den Boot-Manager ansprechen zu können, müssen Sie noch Ihre Betriebssystemlaufwerke mit Hilfe des Menüpunktes In Menü aufnehmen dem Boot-Manager zuordnen.

6. Nun verlassen Sie den FDISK mit Sichern und Ende .

7. Nach erfolgtem neuen Systemstart wird das System, sofern alles geklappt hat, als erstes beim Menü des Boot-Managers ankommen.

Ich hoffe, daß alles geklappt hat und Sie zufrieden sind.

Der Boot-Manager kann nicht nur aus dem FDISK-Menü heraus, sondern auch befehlsgesteuert beeinflußt werden:

2.2.2 ÜBUNG: Arbeit mit dem Dienstprogramm "setboot"

Sofern Sie den *Boot-Manager* installiert haben, kann dieser nicht nur mithilfe des FDISK beeinflußt werden, sondern auch mittels Befehlen, die in einer Befehlszeile eingegeben werden.

Ziel der Übung

Sie lernen die Möglichkeiten kennen, den Bootmanager nachträglich und jederzeit zu beeinflussen.

Durchführung

1. Rufen Sie das Dienstprogramm *setboot* auf.
   - Beginnen Sie indem Sie eine OS/2-Befehlszeile öffnen.
   - Geben Sie ein:

   ```
   setboot /t:100
   setboot /b
   ```

2. Mit dem ersten Befehl wird die Wartezeit bis zum Starten der letzten Betriebssystempartition auf 100 Sekunden gesetzt.
3. Mit dem zweiten Befehl wird Ihr System abgeschlossen und erneut „gebootet".

2.2.3 ÜBUNG: Erweiterter Modus des Bootmanagers

In der ersten Übung für den *Boot-Manager* wurde Ihnen jeder Schritt mitgeteilt. Sie haben jetzt die Gelegenheit, selbst etwas herauszufinden.

Ziel der Übung

Sie bekommen ein paar Aufgaben gestellt. Finden Sie bitte heraus, wie die Lösungen aussehen.

Durchführung

1. Wie können Sie Ihren *Boot-Manager* so steuern, daß er beim Hochfahren des Systems übergangen wird?

   Tragen Sie Ihre Lösung bitte hier ein.

   - ______________________________

2. Geben Sie bitte an, wie Sie den *Boot-Manager* so steuern können, daß er sich im Erweiterten Modus darstellt.

   - ______________________________

3. Was bedeutet der Erweiterte Modus?

   - ______________________________

Durch den Boot-Manager wird es möglich, mehrere **Primäre Partitionen** auf einer Festplatte anzulegen. OS/2 V2 kann in einer **Erweiterten Partition liegen**. Das macht das nachträgliche Installieren unter Umständen einfacher.

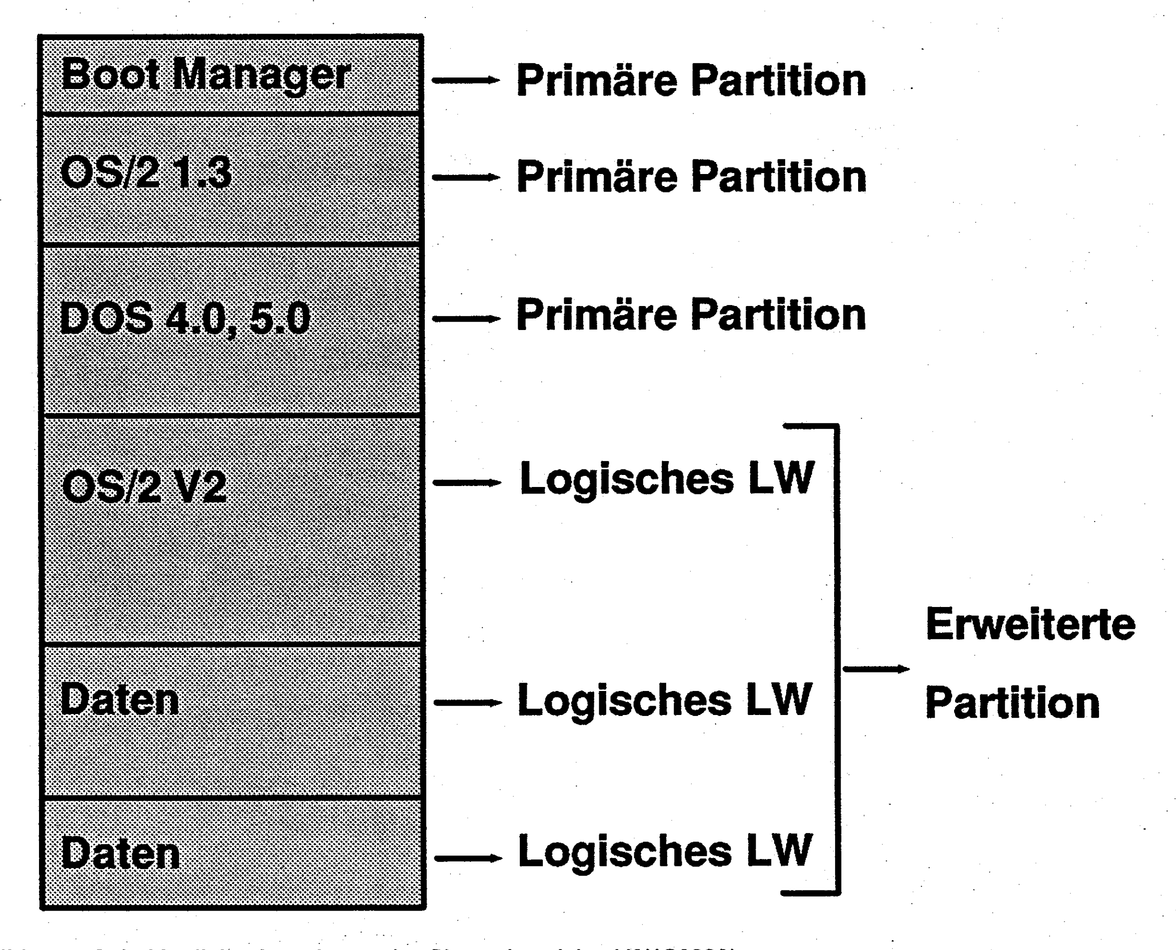


Abbildung 2-4. Mögliche Anordnung der Plattenbereiche (J8HG0220)

Logische Laufwerke

Um mehrere Betriebssysteme, die alle auf die primäre Partition „C“ angewiesen sind, über den Bootmanager starten zu können, ist es möglich, einen Plattenbereich samt logischem Laufwerk als *hidden* festzulegen.

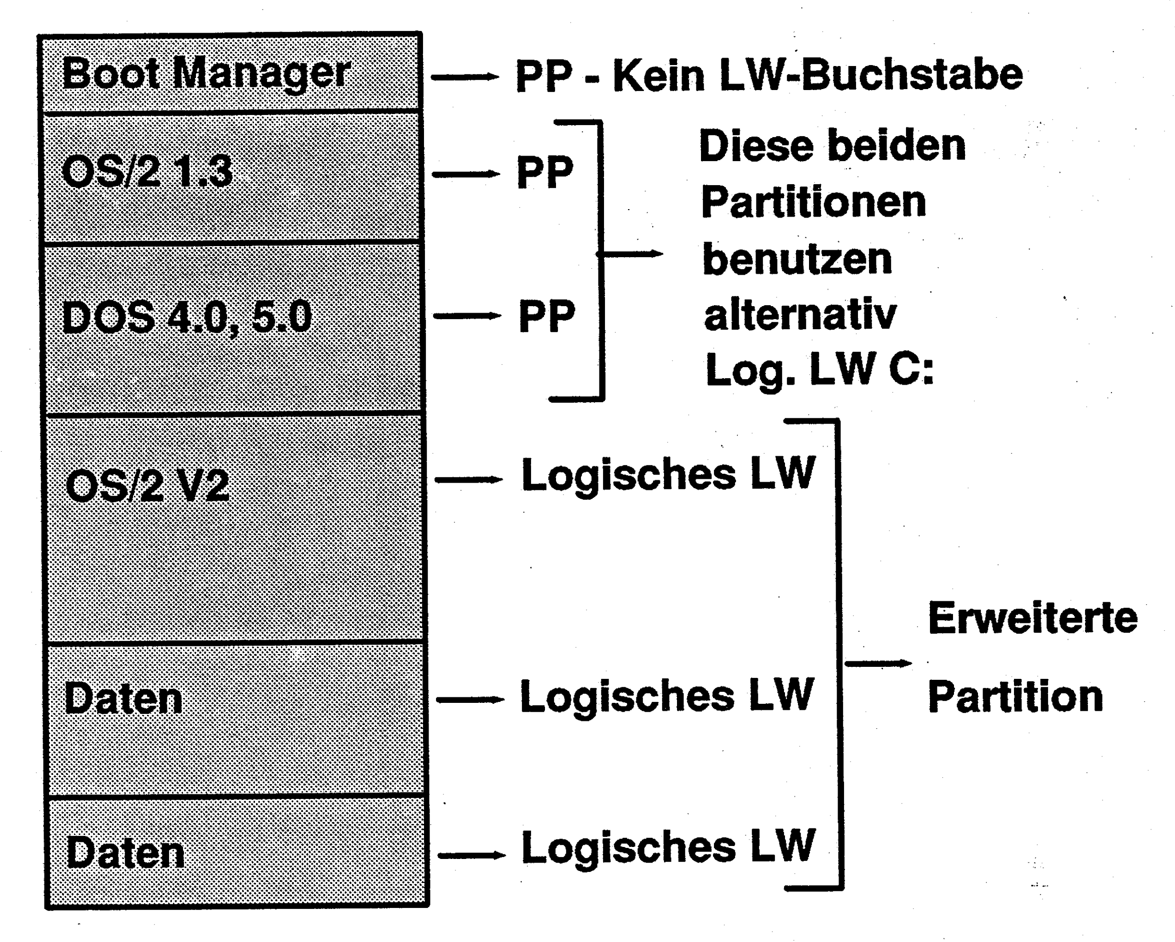


Abbildung 2-5. Verwendung Bezeichnung Logischer Laufwerke 1 (J8HG0225)

Probleme bei der Benutzung der Laufwerksbuchstaben könnnen sich ergeben, wenn unterschiedliche Dateisysteme (FAT und HPFS) verwendet werden. Dadurch können sich Verschiebungen bei den Zuordnungen ergeben, welche zu mancherlei Problemen führen können. Z.B. können Windows-Programme sowohl unter DOS und Windows wie auch unter OS/2 V2 laufen. Die Pfadangaben können in .INI-Dateien stehen und sind dann nur in einem Falle brauchbar.

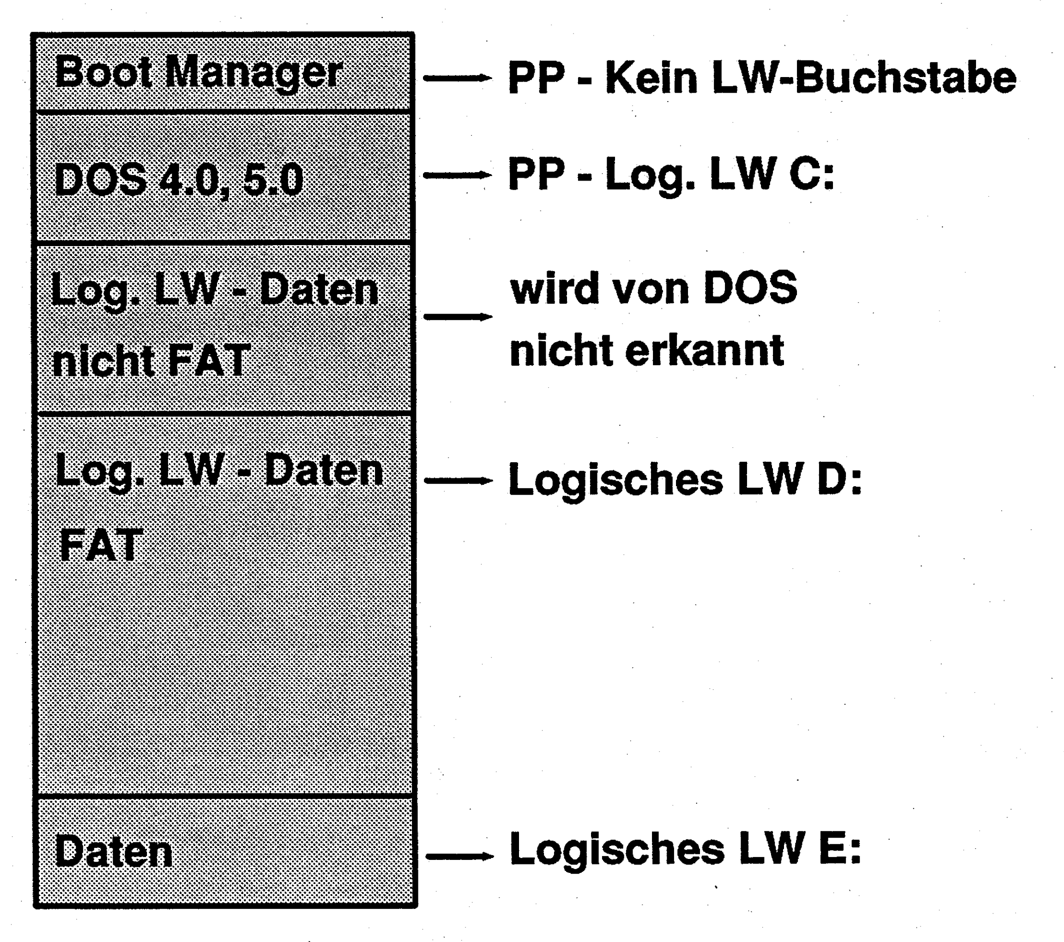


Abbildung 2-6. Verwendung Bezeichnung Logischer Laufwerke 2 (J8HG0230)

Empfehlung

In der CONFIG.SYS unter DOS sollte der LASTDRIVE-Parameter Platz für mögliche Logische Laufwerke bei Verwendung eines Netzwerk-Server lassen; z.B. LASTDRIVE= H.

Besondere Beachtung muß der Laufwerkszuordnung geschenkt werden, wenn auf einer weiteren Festplatte ebenfalls Primäre Plattenbereiche angelegt werden.

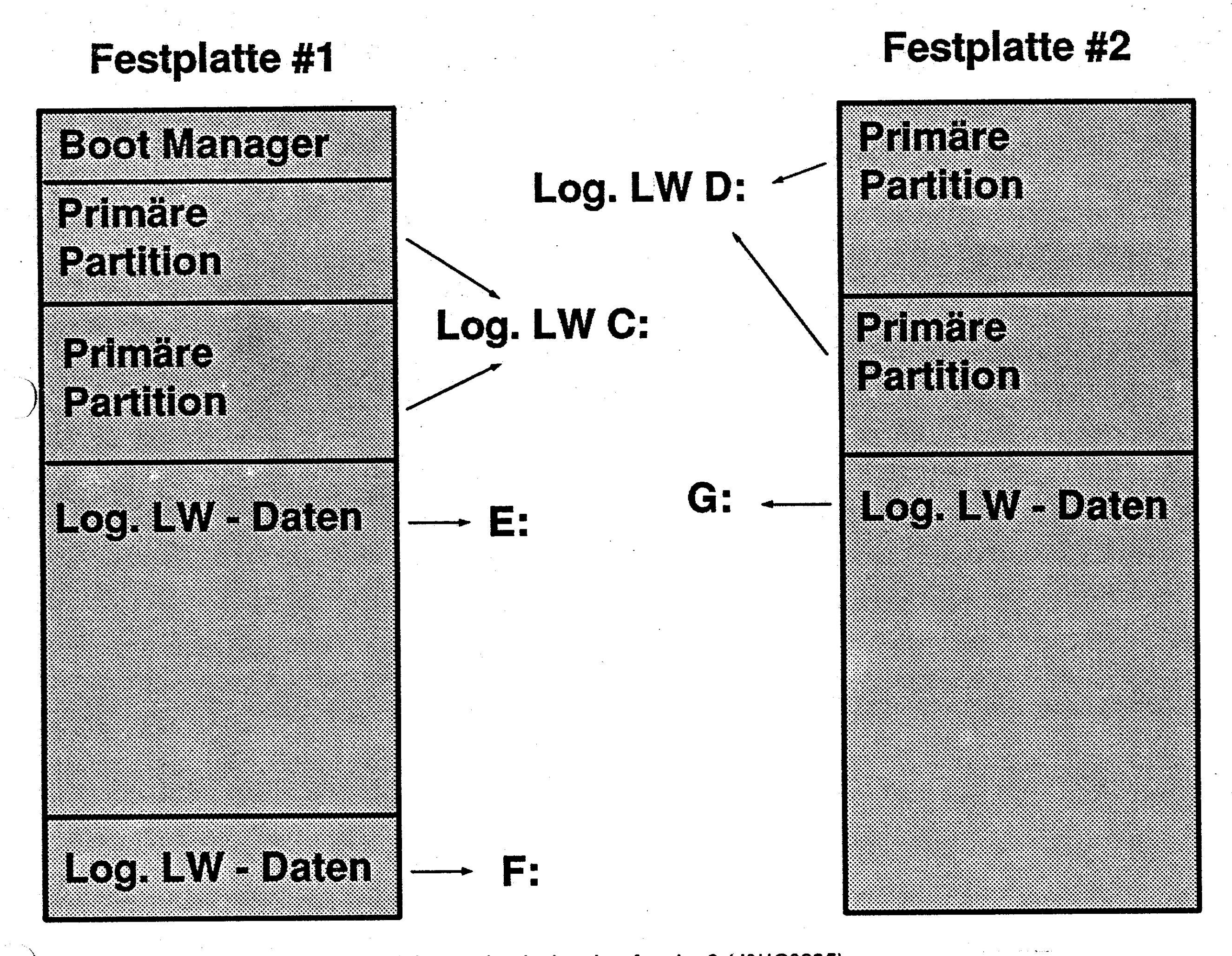


Abbildung 2-7. Verwendung Bezeichnung Logischer Laufwerke 3 (J8HG0235)

Empfehlung

Auf einer zweiten Festplatte sollte keine Primäre Partition sein.

2.3 Installation anderer Betriebssysteme

Nach der Installation des Boot-Manager muß nicht sofort OS/2 Version 2 installiert werden. Es kann sogar sinnvoll sein, andere Betriebssysteme und Programme zuerst zu laden. Dadurch ist es möglich, daß OS/2 bei der Installation andere Programme erkennt und gleich die richtigen Arbeitsordner aufbaut.

DUAL-BOOT

Es gibt, wie auch unter OS/2 VERSION 1.3, die Möglichkeit, die Funktion DUAL-BOOT zu installieren. Während unter OS/2 VERSION 1.3. DUAL-BOOT eine häufig genutzte Möglichkeit darstellte, eine DOS-Anwendung, die auf eine bestimmte DOS-Version angewiesen war, laufen zu lassen, ist unter OS/2 VERSION 2.0 dies nicht mehr erforderlich.

In OS/2 VERSION 2.0 gibt es eine Vielzahl von Möglichkeiten eine Virtuelle DOS Maschine den Anforderungen einer Anwendung anzupassen.

1. Anpassen einer Virtuellen DOS Maschine mit verschiedenen Parametern.
2. Booten einer originalen DOS-VERSION in einer Virtuellen DOS Maschine.
3. Sogar der Zugriff auf HPFS-formatierte Festplatten ist möglich.

Daher gibt es heute nur noch einen Grund, DUAL-BOOT zu installieren. Wenn Sie eine DOS-Anwendung betreiben wollen, die *ill behaved* ist, also direkt auf die Hardware zugreift, kann es unter OS/2 zu Schwierigkeiten kommen, da sich das Multitasking-Betriebssystem selbstverständlich die Verwaltung des gesamten Systems vorbehält.

Besonders zu beachten sind bei DUAL-BOOT:

1. DUAL-BOOT setzt die Installation beider Betriebssysteme in EINER **Primären Partition** auf der ersten physischen Platte voraus.

2. Wenn es sich bei dem verwendeten DOS bei Versionen bis 3.30 handelt, müssen diese in den ersten 32 MB der Festplatte installiert werden.

3. Das jeweilige DOS sollte vor OS/2 installiert worden sein.

4. Wenn Sie außerdem den **Boot-Manager** installieren, müssen Sie ihn inaktivieren und die DUAL-BOOT Partition als Startpartition definieren.

5. Wenn Sie aus OS/2 heraus DUAL-BOOT durchführen, verlassen Sie es komplett, d.h. es wird ein Systemabschluß durchgeführt und mit *native* DOS neu gestartet.

6. Da DOS dann völlig unabhängig von OS/2 gestartet wird, können Sie von dort aus nicht auf HPFS-Platten zugreifen.

7. Ihre DOS-autoexec.bat muß editiert werden, bei DOS 5.0 z.B.:

```
set comspec=c:\dos\command.com
path c:\dos
append=c:\dos
copy c:\dos\command c:\ >nul
```

8. Ihre DOS-config.sys muß editiert werden:

```
shell=c:\dos\command.com /p
```

9. DUAL-BOOT wird mit folgendem Befehl durchgeführt:

   - um von OS/2 nach DOS zu booten

     ```
     BOOT /DOS
     ```

   - um von DOS nach OS/2 zurück zu booten

     ```
     BOOT /OS2
     ```

Die Pfadangaben für die COMMAND.COM sind erforderlich, da die Datei aus dem ROOT-Verzeichnis gelöscht wird und daher sonst nicht gefunden werden könnte.

Empfehlung für die Vorgehensweise für eine Installation:

1. Installation Boot-Manager,
2. Bereich *(Partition)* für z.B. DOS auf Installierbar setzen,
3. DOS installieren,
4. DOS-Programme installieren
5. Bei Bedarf Windows und Windows-Programme installieren,
6. Mit Installationsdisketten von OS/2 booten
7. OS/2 Version 2-Systembereich als Installierbar markieren,
8. OS/2 Version 2 installieren,
9. Am Ende der Installation können die Oberflächen für Windows und DOS vom OS/2 erstellt werden.

Installation unter Verwendung einer Antwort-Datei, einer

sogenannten *response file*. Während einer Installation erhalten Sie einige Bildschirme, die Entscheidungen von Ihnen erwarten.

Für den Fall, daß Sie mehrere Systeme gleicher Konfiguration installieren möchten, können Sie sich auf der Diskette 1 des Diskettensatzes die Möglichkeit dafür schaffen. Das hat den Vorteil, das Sie wärend der Installation keine Fragen mehr beantworten müssen.

Wir stellen Ihnen hier den Ablauf dar, damit Sie evtl. am Arbeitsplatz darauf zurückgreifen können.

1. Kopieren Sie die Diskette 1.
2. Nach der Installation befindet sich die ASCII-Datei: „SAMPLE.RSP" im Verzeichnis OS2\INSTALL.
3. Kopieren Sie diese unter dem Namen „OS2SE20.RSP"
4. Editieren Sie sie entsprechend Ihrer Wünsche unter Verwendung z.B. des OS/2-Systemeditors.
5. Die Datei enthält für jeden erforderlichen Eintrag eine Erklärung. Es ist aus Platzgründen zu empfehlen, während der Bearbeitung der Datei die nicht mehr benötigten Bemerkungen zu löschen.
6. Kopieren Sie weiterhin die Datei „RSPINST.EXE" aus dem Verzeichnis auf Ihre neue Diskette 1.
7. Den eigentlichen Installationsprozeß beginnen Sie wie gewohnt mit der Installationsdiskette und legen Sie danach Ihre neue Diskette 1 ein.
8. Sie werden jetzt nur noch zum Diskettenwechsel aufgefordert, müssen aber nicht mehr Entscheidungen treffen, bzw. Fragen beantworten.

Remote Installation

Es gibt die Möglichkeit, das Betriebs-System auch von einem Server aus zu installieren. In diesem Kurs würde es zu weit führen, alle Facetten zu beleuchten, daher stellen wir Ihnen hier diese Möglichkeit im Überblick dar. Sie werden daran die Komplexität feststellen. Genaueres finden Sie dazu in dem Redbook: **Remote Installation and Maintenance** mit der Dokumenten-Nummer **GG24-3780**. Die nötigen Schritte sind:

1. Installation des IBM OS/2 LAN SERVER V2.0 mit OS/2 Remote IPL support auf der Server-Maschine.
2. Laufenlassen des Dienstprogramms RIPLINST auf dem Server zur Erstell- ung der IPL-Verzeichnisse.
3. Entpacken der CID-Datei aus dem Diskettensatz
4. Sie benötigen hierfür von der Diskette 7 die Dateien:

   ```
   SEIMAGE.EXE
   SEDISK.EXE
   SEMAINT.EXE
   SEINST.EXE
   ```

   Diese finden Sie nicht direkt auf der Diskette und auch nicht im installierten OS/2, sondern sie sind in der gepackten Datei: CID enthalten.
5. Sie müssen mit dem Dienstprogramm UNPACK entpackt werden.

   ```
   UNPACK A:CID
   ```

6. Die o.g. Dateien werden dann in Ihr Verzeichnis OS/2\INSTALL auf der Festplatte gestellt. Sie können selbstverständlich auch einen Zielpfad mit angeben, z.B.:

   ```
   UNPACK A:CID C:\TEMP
   ```

7. Das Dienstprogramm SEIMAGE.EXE erstellt Ihnen auf Ihrer Festplatte eine Verzeichnisstruktur, die alle Disketten des Installationssatzes beinhaltet. Wichtig ist, daß Ihnen bewußt ist, daß es sich dabei nicht um Kopien der Disketten als solcher handelt, sondern um Disketten - *Images*, also Disketten-Abbilder.

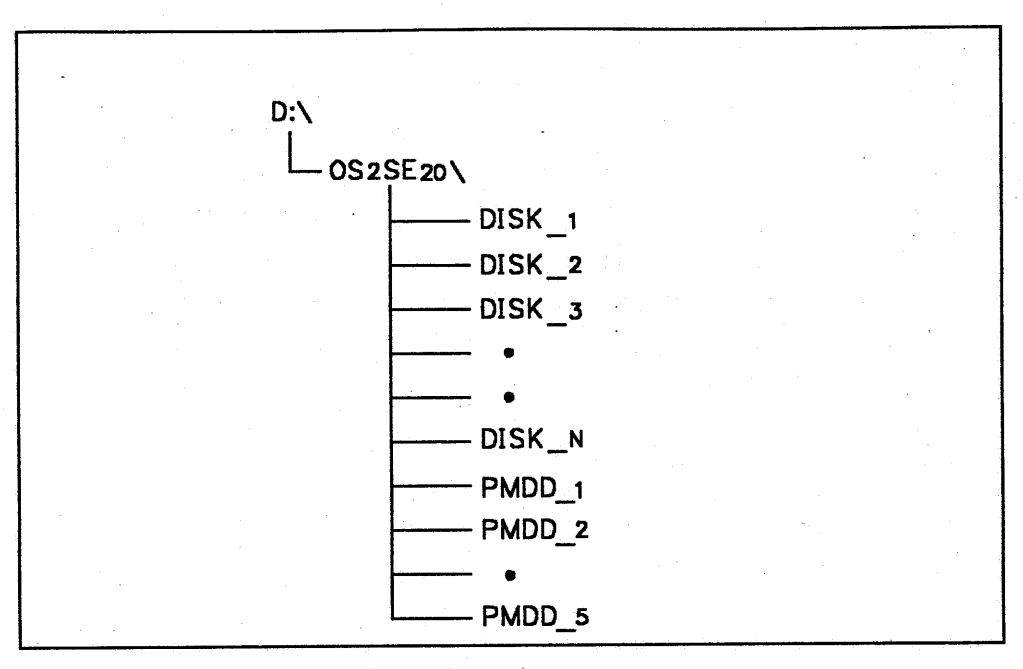


Abbildung 2-8. SEIMAGE Verzeichnis Struktur (J8HG0236)

Der erforderliche Aufruf ist:

```
SEIMAGE /S:A: /T:D:\OS2SE20
```

Dies hat zur Folge, daß die *source* (/S) Laufwerk A: ist, das *target* (/T) Laufwerk D:. Das Zielverzeichnis heißt OS2SE20.

8. Im nächsten Schritt wird das Dienstprogramm SEDISK verwendet. Diese erstellt Ihnen die Diskette 0, also die Installationsdiskette und die Diskette 1.

   - Sie benötigen dafür zwei formatierte Disketten.
   - Die Disketten-Images müssen vorher auf der Festplatte erstellt worden sein.
   - Die Eingabe ist:

     ```
     SEDISK /S:D:\OS2SE20 /T:A:
     ```

   - Hiermit werden auf Laufwerk A: die Boot-Disketten erstellt.
   - Diese Disketten beinhalten nicht die LAN resp. die Novell Treiber. Diese Anpassungen müssen noch manuell vorgenommen werden.

9. Das Dienstprogramm SEMAINT wird nur für upgrades von einer Server-Maschine benötigt.

10. Das Dienstprogramm SEINST nimmt die Installation auf der Target- Maschine vor.

11. Einrichten des Remote IPL Konfiguration Prozesses mit dem Dienst- programm GETRPL auf dem Server.

12. Erstellung einer MASTER WORKSTATION DEFINITION.

13. Erstellung der RESPONSE FILES für die *client*-Maschinen.

14. Erstellung der FILE INDEX TABLE (FIT) und der CONFIGRI.20.

15. Erweiterung der MASTER WORKSTATION DEFINITIONS durch NAME und LAN ADAPTER Adressen.

16. Anpassung der RPL.MAP Datei.

17. Start des REMOTEBOOT auf dem Server.

18. Erstellung der Installation Diskette mit RPLDISK oder RPLENABLE.EXE.

19. Transfer zu den *clients*.

20. Installationsdiskette in die *clients*.

21. Eigentliche Installation auf den *clients*..

22. *Reboot* der *clients*.

Mittels einer speziellen CID-Software gibt es jetzt auch die Möglichkeit, *Remote Installation* menügesteuert durchzuführen.

Kapitel 3. OS/2 Version 2, Architektur HW

Ziele dieses Kapitels

- Dieses Kapitel zeigt einige der Architekturgrundlagen von PC und PS/2.
- Sie verstehen die auf der Hardware-Seite zu erfüllenden Voraussetzungen.

Inhalt dieses Kapitels

3.1 Architektur Hardware 3-3
3.1.1 Entwicklung der Mikroprozessoren 3-4
3.1.2 Prozessor Architektur 3-6
3.1.3 Die Prozessorfamilie Intel 80x86 / 80x87 3-8
3.1.4 80286 Architektur 3-8
3.1.5 Der Prozessor 80386 3-9
3.1.6 80386 Architektur 3-9
3.1.7 Der Prozessor 80386SX 3-12
3.1.8 Der Prozessor 80386SLC 3-12
3.1.9 Der Coprozessor 80387 3-13
3.1.10 80486 (i486) Mikroprozessor 3-14
3.1.11 Der Prozessor 80486SX 3-15
3.1.12 Der Coprozessor 80487 3-15
3.1.13 Pentium 3-16
3.1.14 Overdrive Prozessor 3-17
3.1.15 Taktrate 3-18
3.1.16 Was ist ein Prozessor 80486DX2 66? 3-18
3.2 Architektur Mikrokanal und Implementation 3-19
3.3 Mikrokanal Architektur 3-20
3.3.1 PS/2 BUS Architektur 3-22
3.4 BUS Architektur des Personal Computer 3-26
3.5 System Konfiguration 3-28
3.6 Hardware als Einflußfaktor 3-31

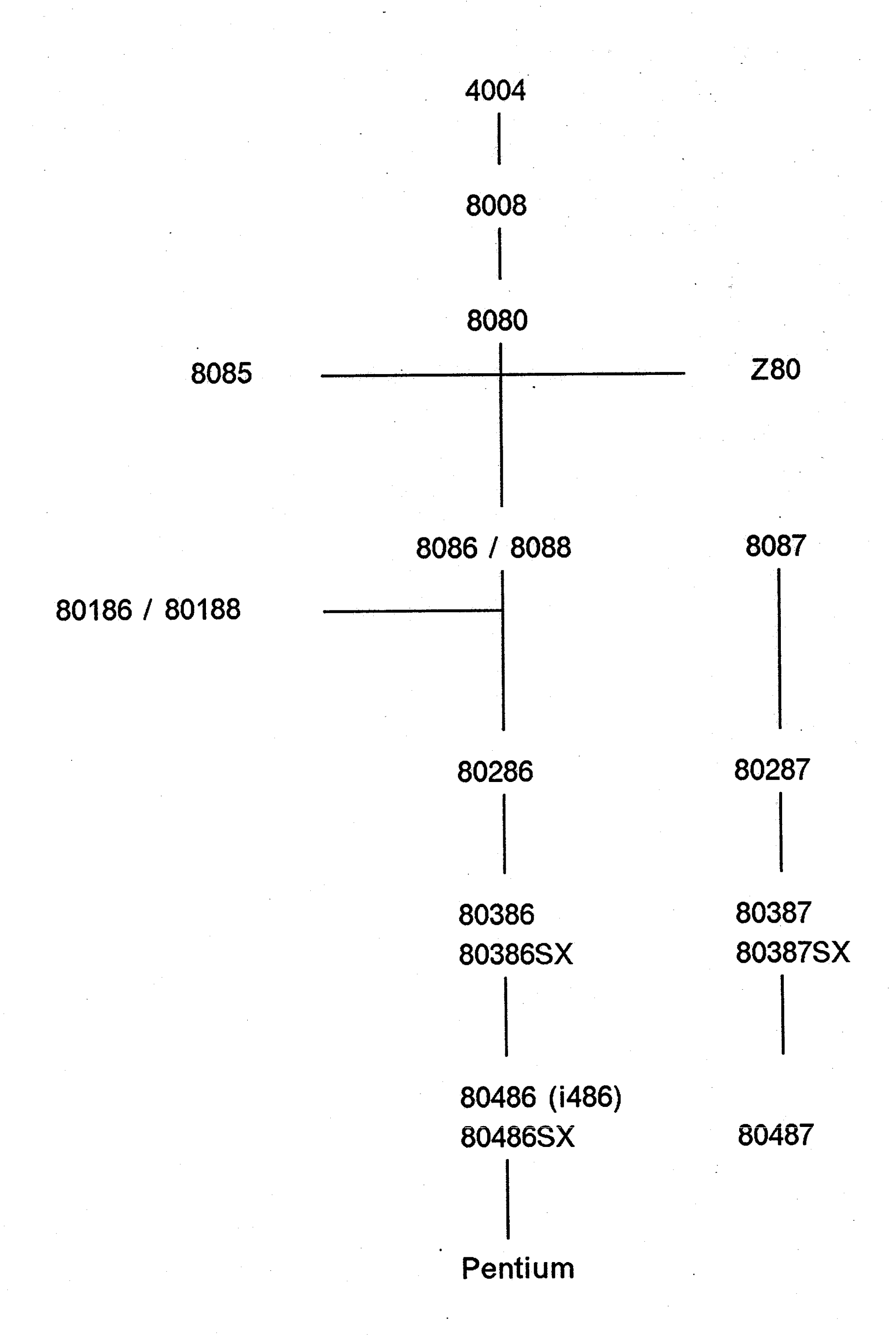
3.1 Architektur Hardware

Ziele dieses Kapitels

- Am Ende dieses Kapitels erkennen Sie einige Einschränkungen der PC/DOS-Welt.
- Sie wissen, welche Prozessoren z.Zt. OS/2 unterstützen und welche wesentlichen Eigenschaften die einzelnen besitzen.

Der Inhalt dieses Kapitels dient lediglich der Klarstellung von Begriffen oder Fehlinformationen, er soll Sie n i c h t zum „Spezialisten in Hardware-Fragen" machen!

3.1.1 Entwicklung der Mikroprozessoren



Wenn in diesem Kapitel zu den 80x86 Prozessoren kein Zusatz, wie SX o.ä. genannt wird, dann ist das gleichbedeutend mit sog. DX Prozessoren; ein 80386 ist also das gleiche wie ein 80386DX.

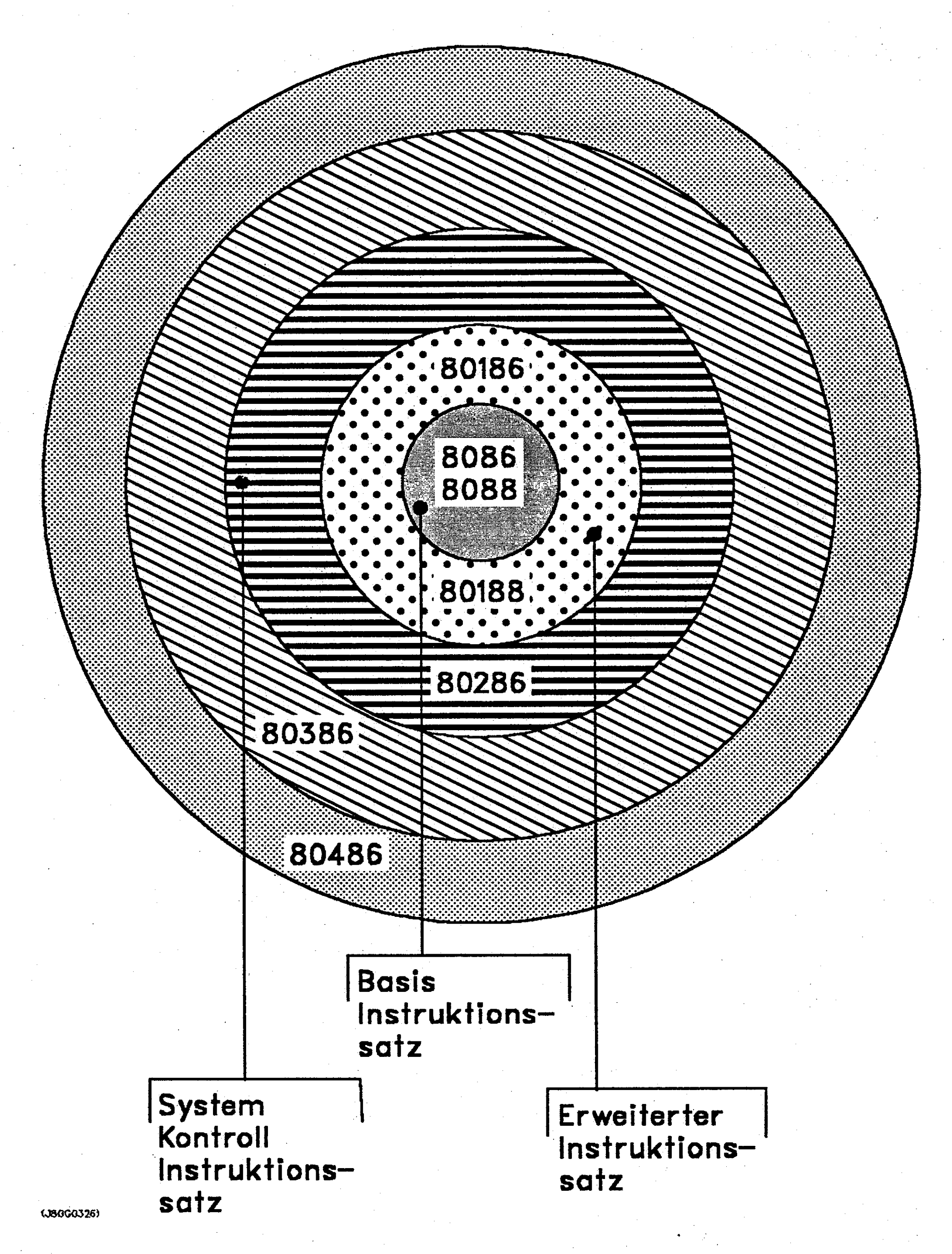
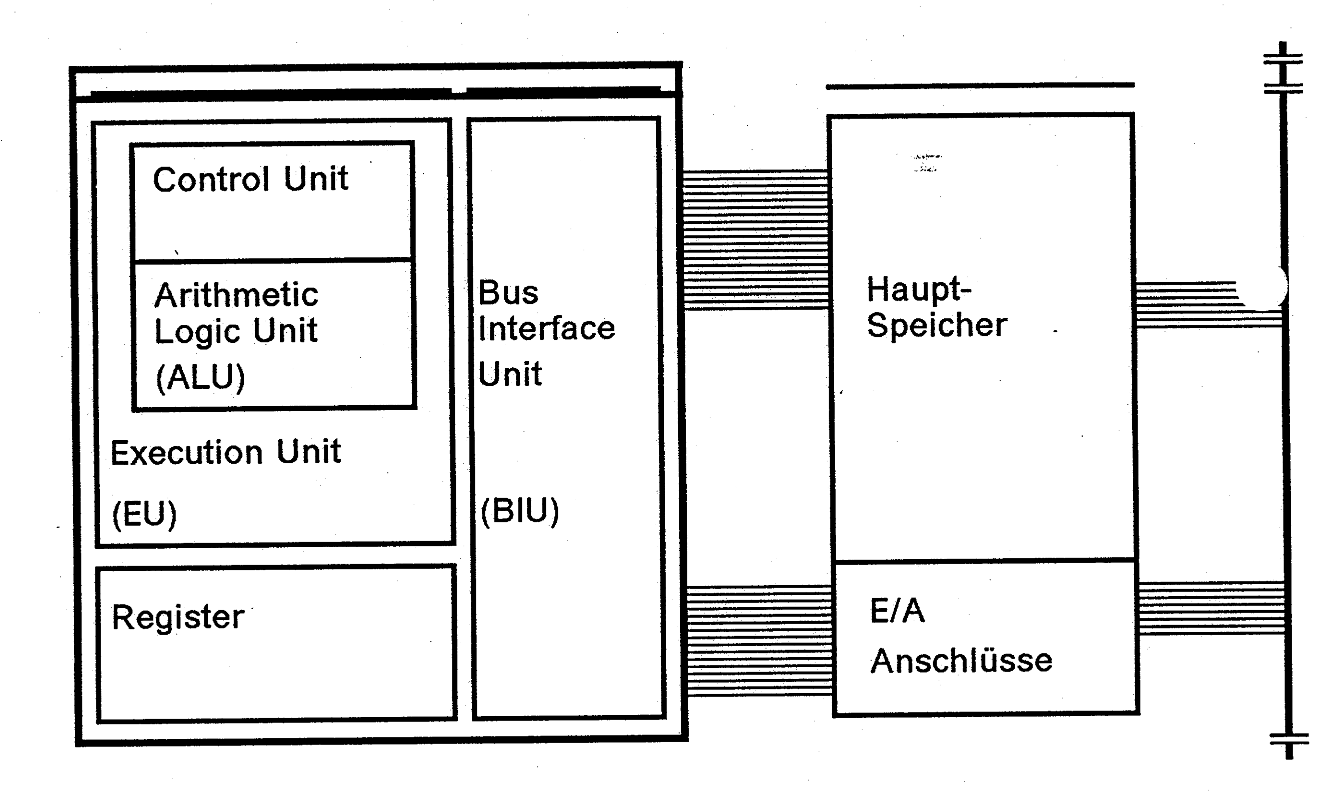


Abbildung 3-1. Instruktions-Hierarchie Intel-Prozessoren (J8HG9901)

3.1.2 Prozessor Architektur

Alle bekannten INTEL-Prozessoren basieren auf den gleichen Architektur- Vorraussetzungen. Daher soll hier grundlegend darauf eingegangen werden.



Systemplatine

- Die Systemplatine besteht aus 5 funktionellen Bereichen:
  - Prozessor,
  - ROM,
  - RAM,
  - I/O-Adaptern und
  - I/O-Kanal bzw. BUS

Register

Bestimmte Speicherstellen im Prozessor werden als **Register** bezeichnet.

Sie können

- Daten,
- Adressen oder
- Steuerinformationen

speichern. Sie befinden sich im Prozessor, daher hat er auf sie einen schnellen Zugriff.

Die Größe der Register ist unterschiedlich bei den Prozessoren. So besitzt z.B. der 80286 16-bit Register, während der 80386 32-bit Register (plus die 16-bit Register des 80286) enthält.

3.1.3 Die Prozessorfamilie Intel 80x86 / 80x87

3.1.4 80286 Architektur

1. Bis 16 MB Physikalischer Hauptspeicher adressierbar.
2. Bis 1 GB virtueller Speicher / TASK möglich.
3. Multitasking-fähig
4. Segmentgrößen von 1 Byte bis 64 KB (keine 16-Byte-Boundary wie 8088/86)
5. 4 eigenständige Einheiten
   - BUS
   - Instruction
   - Execution
   - Adressierung
6. Globale Segmente, Lokale Segmente
7. Privilegierte Ebenen
8. Operations Modi
   - REAL Modus
   - Protected Modus

3.1.5 Der Prozessor 80386

Er arbeitet sowohl intern, als auch extern mit 32-bit Leitungen, überträgt also beispielsweise immer 32 Bits gleichzeitig parallel. Sein Vorgänger - der 80286 - war lediglich ein 16-bit Prozessor. Er benötigt natürlich eine entsprechende Hauptplatine mit den kompatiblen Adapterkarten. Auf dieser Hauptplatine befindet sich ein leerer Steckplatz für einen mathematischen Coprozessor Er und sein Nachfolger - der 80486 - sind z. Zt. die am weitesten verbreiteten. Ein Betriebssystem für die Zukunft kann also beruhigt einen solchen als Minimum voraussetzen.

3.1.6 80386 Architektur

80386 Potential

4GB

TASK 1 | TASK 2 | TASK n

0

- Große Programme und Datenmengen bis zu 4 GB
- 17 Mikrosekunden Task Umschaltung (60000 mal/Sekunde)
- 1 Segment pro Programm

Charakteristika

1. 4 GB Physischer Hauptspeicher (2^{32})
2. 64 Terabyte Virtueller Speicher / TASK (2^{46})
3. 32 Bit Adressleitungen
4. 32 Bit Datenleitungen (Lesen/Schreiben)
5. Multitasking- und Multiprocessing-fähig
6. Segmentgrößen von 1 Byte bis 4 GB
7. Paging Einheit 4 KB
8. 6 eigenständige Einheiten
   - BUS
   - Code
   - Instruction
   - Execution
   - Segment
   - Paging
9. Interne Pipeline-Architektur
   - 6 verschiedene Instruktionen in unterschiedlichen Ausführungsstufen
10. 3 Operations Modi
    - REAL Modus (1 MB)
    - Protected Modus
    - virtueller 8086 Modus
11. HW Fixed-Point Multiply und Divide

REAL Modus

REAL Modus wird so genannt, weil er einem Anwendungsprogramm die Nutzung des realen (physischen) Hauptspeichers unterhalb 1 MB erlaubt. Einschränkend muß gesagt werden, daß der REAL Modus nur ein schneller 8086 ist, also auch nur ein Hauptspeicherbereich bis 640KB genutzt werden kann. Keine der eigentlichen Stärken des 80286 wird hierbei unterstützt, d.h. kein Speicherschutz, kein Multitasking.

1. Technik läßt Nutzung höherer Taktraten zu
2. 8086/88 Instruktionssatz + Erweiterten Befehlssatz (80188/186)
3. Hardware Multiplikationsfunktionen

Protected Modus

Ein Schutzsystem verhindert im Protected Modus, daß Anwendungsprogramme **direkt** physischen Hauptspeicher adressieren können. Dieses ist wichtig bei Multiprogramming, für das der 80286 entwickelt worden ist. Der 80286 bietet:

1. Unterstützung neuer Funktionen, wie
   - Virtuelle Speicherverwaltung
   - Erweiterte Adressierung (16MB)
   - Multitasking
   - Schnelle Taskumschaltung
   - Sicheres Speicherschutzsystem
2. REAL MODE Programme sind in diesem Modus nicht lauffähig.
3. OS/2 benutzt diesen Modus

Virtueller REAL Modus

Dieser Modus wird erst ab dem Prozessor 80386 unterstützt. Was passiert dabei?

- Der einer **Virtuellen DOS-Maschine** zugeordnete Speicherbereich wird nach DOS-Erfordernissen adressiert

- Darin wird der 8086-Modus emuliert
- Mehrere DOS-Sitzungen sind möglich
  - Sie sind „Gäste" des OS/2
- Verwaltung durch OS/2
- Außen herum läuft OS/2 mit seinem **Protected Modus**
- Auch eine DOS-Anwendung nimmt dadurch voll am **Multitasking** und dem Schutzkonzept des Betriebssystems.
- 80386 Fehler-Behandlungsroutine
- E/A werden vom OS/2 abgearbeitet
- Anwendungen im 80x86- und virtuellen Modus laufen gleichzeitig

3.1.7 Der Prozessor 80386SX

Er ist intern ein vollwertiger 32-bit Prozessor und entspricht dem 80386. Extern allerdings arbeitet er mit 16-bit Leitungen, das ermöglicht es den PC-Herstellern, Hauptplatinen und Adapterkarten zu verwenden, die noch aus der Zeit des 80286 stammen. Er wird allerdings genauso wie der 80386 durch OS/2 unterstützt.

3.1.8 Der Prozessor 80386SLC

Der 80386SLC ist eine Entwicklung der IBM, er entspricht einem 80386SX mit einem internen Cache. Ein Prozessorcache ist ein Zwischenbereich, in den vom Hauptspeicher Daten bzw. Instruktionen „vorgeladen" werden. Dann muß der Prozessor nicht auf die relativ langsame Zugriffszeit des Hauptspeichers (ca. 60-80 nsec) warten und findet die benötigten Informationen/Instruktionen schneller vor. Ihm wird dann gewissermaßen „zugearbeitet"

3.1.9 Der Coprozessor 80387

Er beschleunigt Gleitkommarechnungen sehr stark, so daß diesbezügliche Anwendungen oder Programme entsprechend sehr viel schneller laufen - beispielsweise Tabellenkalkulationen etc.

3.1.10 80486 (i486) Mikroprozessor

Der 80486 besteht im wesentlichen aus 4 Komponenten:

1. Prozessoreinheit ist 80386-kompatibel
2. 80387-kompatibler Arithmetikeinheit
3. Memory-CACHE Controller mit
4. Integriertem 8 KB CACHE-Speicher

Durch Einsatz von RISC-Technik hat der i486 bei gleicher Taktfrequenz wie der 80386 (25 MHz) ca. 3-fache Leistung gegenüber dem 80386.

Der integrierte mathematische Co-Prozessor ermöglicht es, Rechenoperationen bis zu 100 mal schneller durchzuführen.

Die neue Adressierungstechnik erlaubt es, auf externen Speicher durchschnittlich dreimal so schnell zuzugreifen.

z.B.: Wert aus Speicher lesen:

- 80386 = 4 Takte
- 80486 = 1 Takt

z.B.: Unterprogrammaufruf:

- 80386 = 9 Takte
- 80486 = 3 Takte

Aus der Sicht des Programmierers ändert sich beim 80486 gegenüber dem 80386 kaum etwas. Alle Möglichkeiten des 80386 sind ebenfalls im 80486 enthalten. Er enthält allerdings einige Zusätze. Wichtige, die Leistungsfähigkeit betreffende Erweiterungen sind ein Prozessorcache, wie er schon beim 80386SLC angesprochen wurde, sowie der integrierte mathematische Coprozessor, der beim 80386 als 80387 dazugesteckt werden konnte. Diese beiden Dinge sind im 80486 also bereits enthalten.

3.1.11 Der Prozessor 80486SX

Dieser Prozessor ist ein vollwertiger 80486, dem nur eines fehlt: er hat keinen integrierten mathematischen Coprozessor. Ansonsten ist er dem 80486 ebenbürtig.

3.1.12 Der Coprozessor 80487

Wenn Sie einen 80486SX auf der Hauptplatine haben und die Funktion des mathematischen Coprozessors hinzuinstallieren wollen, dann können Sie - wie beim 80387 - einen 80487 auf den freien Sockel stecken. Dieser legt dann allerdings den 80486SX tot und übernimmt die volle Arbeit eines 80486 Prozessors.

3.1.13 Pentium

Dieser Prozessor hat bewußt nicht den Namen „80586“ erhalten. Mit dem Namen **PENTIUM** hat INTEL eine neue Prozessorgeneration eingeleitet.

- Wie alle Prozessoren aus der PC- PS/2 Reihe ist auch dieser kompatibel zu den älteren.
- Die Taktfrequenz wurde erneut gesteigert auf 60/66/100MHz
- Der Prozessor basiert auf einer Breite von 64 bits.
- Er hat je einen Daten- und Instruktions-Cache von je 8 kB.
- parallele Instruktionsverarbeitung
- erweiterter virtueller 8086-Modus. Hierdurch wird unter OS/2 für DOS- und Windows-Programme ein nochmals verbessertes Laufzeitverhalten erreicht.
- Er unterstützt Multiprocessing.
- Auf die Verarbeitungsgeschwindigkeit wirkt sich die **superskalare Architektur** besonders aus.
  - 2 Pipelines, die unabhängig oder auch zusammenarbeiten können.
  - Jede hat ihre eigene **ALU** (arithmetisch logische Einheit)
- Verzweigungsvorhersage, sie steuert die Instruktionen in die jeweils freie Pipeline.
- Hochleistungs-Fließkommaeinheit, oft benötigte Operationen sind bereits hardwaremäßig untergebracht.
- Aufrüstbar mit **OverDrive-Prozessor**

3.1.14 Overdrive Prozessor

Hierbei handelt es sich um die Möglichkeit einen installierten Prozessor aufzuwerten.

- Für den 80486-Prozessor gibt es solchen.
- Er wird in den blauen *upgrade*-Steckplatz neben dem eigentlichen Prozessor eingesteckt.
- Der installierte Prozessor wird in seiner internen Taktfrequenz erhöht. Dies hat den gleichen Effekt wie die Taktverdoppelungs-Technik beim 80486DX2.
  - Dies entspricht dem *double clock*
- Darüber hinaus wird für 1994 ein Overdrive Prozessor (P24T) erwartet, der einen 80486DX2-Prozessor dem Pentium-Niveau näher bringt.

3.1.15 Taktrate

Die Taktrate gibt in Megahertz (MHz) die Geschwindigkeit des Prozessors an. Ein 80386 ist z.B. mit 20, 25 oder 33 MHz getaktet, ein i486 ist meistens mit 33 MHz getaktet. Die Taktrate ist physikalisch begrenzt. Zum einen wird ein sehr hoch getakteter Prozessor irgendwann zum Sender, was natürlich das Umfeld beeinflußt und daher Abschirmungsprobleme bringt. Zum anderen entsteht bei höherer Taktrate mehr Wärme, die die Kühlung des Prozessors aufwendiger macht.

Es gibt auch Prozessoren, die mit ...DX2 bezeichnet werden. Die dahinter stehende Taktrate gilt dann intern, nach außen kommuniziert er mit der halben Taktrate. Das ermöglicht den Einsatz preiswerter Hauptplatinen und Adapterkarten.

Einer Taktrate von 25 MHz entspricht eine Zugriffszeit von 40 nsec (Nanosekunden), eine von 50 MHz entspricht 20 nsec usw.

3.1.16 Was ist ein Prozessor 80486DX2 66?

Ein 80486DX2 66 ist daher nach den vorherigen Aussagen ganz einfach zu entschlüsseln:

- 80486DX heißt, daß es sich um einen vollwertigen i486 handelt.
- 66 entspricht der Taktrate. Er ist also mit 66 MHz getaktet.
- DX2 bedeutet nun, daß der Prozessor nur zur Hälfte mit 66 MHz getaktet ist, nämlich intern. Nach außen verhält er sich auf der Hauptplatine genauso wie ein 80486DX 33 und ist also nach außen (Übertragung etc.) mit 33 MHz getaktet.

Das war doch gar nicht so schwer zu entschlüsseln, oder?

3.2 Architektur Mikrokanal und Implementation

IBM hat mit dem Personal System/2 ein System entwickelt, das nicht nur aus neuer Hardware besteht, sondern aus Komponenten im Bereich der Hardware und der Software, die aufeinander abgestimmt sind.

Der Mittelpunkt dieser Architektur ist der **Mikrokanal**. Eine Computer-Architektur ist eigentlich ein abstraktes Konzept, in dem Regeln definiert sind, die aus augenblicklicher Sicht manchmal wenig Sinn ergeben und das Gerät nur verteuern.

Die IBM hat das PS/2 entwickelt im Hinblick auf die Zukunft, sodaß Erweiterungen in Hard- und Software einfach zu implementieren sind, d.h. die Architektur ermöglicht wesentlich mehr als z.Zt. implementiert ist. **Daher sollte nicht Implementation mit Architektur verwechselt** werden.

Das PS/2 ist als Implementation der Intel 80286 Architektur zu sehen, d.h. eine HW-Realisierung vieler Möglichkeiten des 80286. Bei Betrachtung der Konzeption wird aber schnell sichtbar, daß der 80386 in dieses Konzept eingefügt ist.

IBM hat eine komplette Familie von Mikrocomputern geschaffen, die mit dem Modell 30 den Übergang mit der alten BUS-Version in die neue Mikrokanal-Architektur erlauben.

An die gesamte Palette von Systemeinheiten sind alle E/A-Geräte und Adapter anschließbar, wenn man das Modell 30 ausnimmt. Damit kann vom kleinsten Modell - mit wachsenden Anforderungen des Benutzers - der Ausbau bis zu den leistungsfähigsten Modellen erfolgen. Die Verbindung zu Host's und externen Netzen, sowie internen **LAN's** kann jederzeit hergestellt werden. Frühere Zusätze sind teilweise Standard.

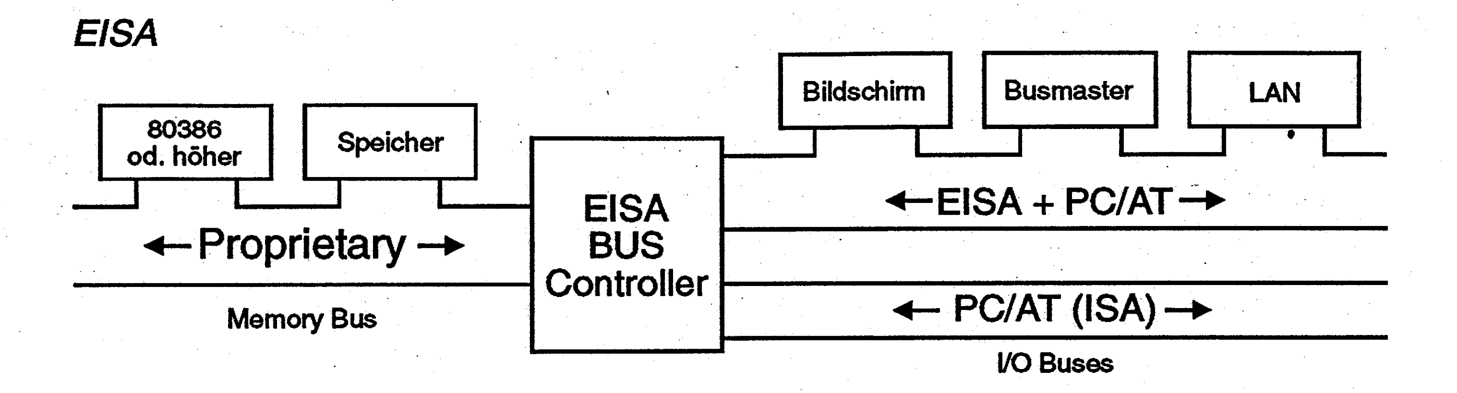
3.3 Mikrokanal Architektur

Wenn der 80x86 Mikroprozessor das Herz des **IBM Personal System/2** ist, dann bildet der BUS die Hauptschlagader. Beim PS/2 wird dieser BUS **Mikrokanal** genannt, weil seine Konzeption sich an der Großsystem-Architektur orientiert. Der Mikrokanal ist für Parallel-Betrieb und *Concurrent Processing* entwickelt worden und bildet die Ausgangsbasis für zukünftige Entwicklungen.

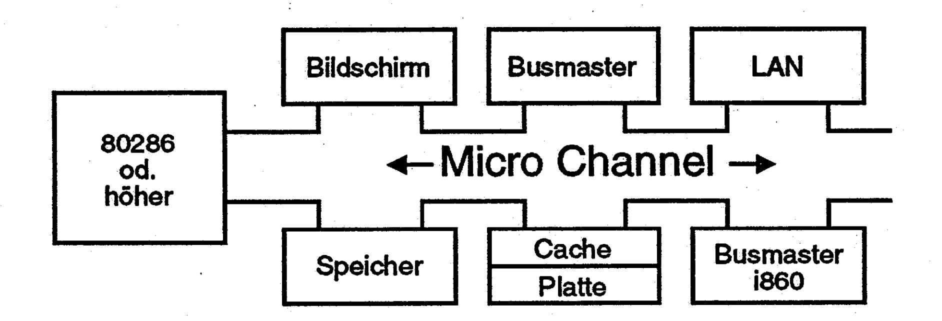
Der BUS ist die Schnittstelle zwischen Systemplatine und Adapter, stellt alle Signale und die Stromversorgung für Parallel-Betrieb und *Concurrent Processing* entwickelt worden und bildet die Ausgangsbasis für zukünftige Entwicklungen. Der BUS ist die Schnittstelle zwischen Systemplatine und Adapter, stellt alle Signale und die Stromversorgung für interne Komponenten und externe Adapter zur Verfügung.

Der Mikrokanal ist **kein** herkömmlicher BUS, sondern setzt sich aus mehreren definierten Signal- und Funktionsgruppen zusammen und bildet ein System- und Prozessor- **unabhängiges Informations-Übertragungssystem**.

EISA



Micro Channel



J8HG2107

Abbildung 3-2. AT-Bus und Mikrokanal (J8HG2107)

3.3.1 PS/2 BUS Architektur

- Grundüberlegung
  - Rasantes Vordringen neuer Technologien
  - Fortschrittliche Zusatzgeräte/Möglichkeiten
- Unterstützung
  - 32-Bit System
  - Großsystemähnliche Funktionen
- Basis für zukünftige Entwicklungen
  - schnellere Verarbeitung
  - gleichzeitige Verarbeitung (Concurrent Processing)
    80286 = bis zu 4 Concurrent Prozessoren
    80386 = bis zu 6 Concurrent Prozessoren
    Mikrokanal = bis zu 15 Concurrent Prozessoren

Konzept

Das neue Konzept der **Mikrokanal** Architektur wird durch folgende Funktionen verwirklicht:

- Programmierbare Optionsauswahl
  Programmable **O**ption **S**elect (POS)
- Ebenen-abhängige, zeitgleiche Unterbrechungen
- Multi-Device Arbitration
- Master Attachment

Nur durch das geänderte Layout in Verbindung mit anderen technischen Änderungen ist eine zuverlässige Funktion unter Einhaltung der gesetzlichen Bestimmungen (FTZ) auch bei weiterem Wachstum möglich.

- Hohe Übertragungsraten / Durchsatzsteigerung
- Keine Abschirmungen
- Allgemeine Betriebserlaubnis (GOP)
- Datenübertragung Asynchron
- Schnelle/Einfache Erweiterungs-/Adapter-Installation
- Neue Controller
- Wachstumspotential, da bei zukünftigen Forderungen anpaßbar

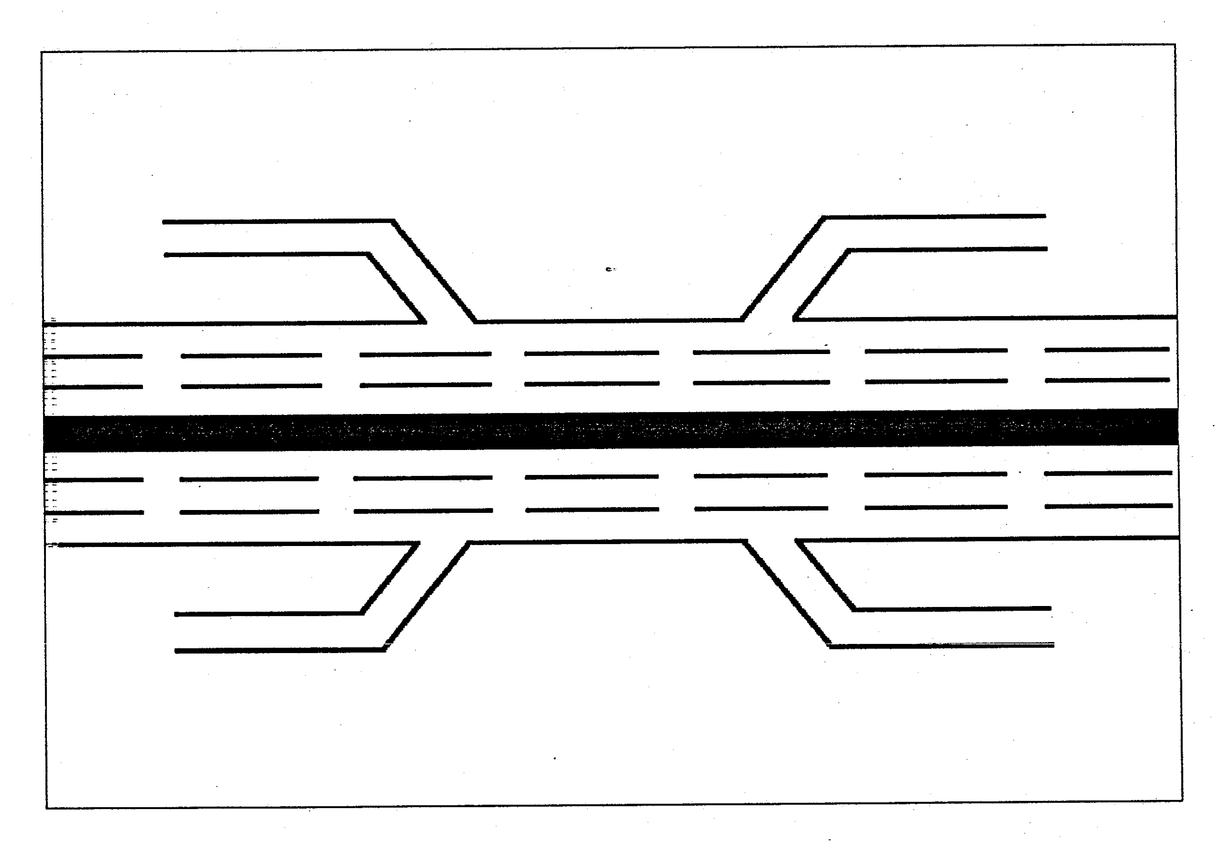
Trotz der „offenen Architektur“: softwarekompatibel zum IBM-Standard bzw. „Industriestandard“.

Signalgruppen

Der **Mikrokanal** wird durch folgende Signalgruppen gebildet:

- Adress-BUS
- Daten-BUS
- Steuerungs-BUS
- Arbitration-BUS
- zusätzliche Unterstützungs-Signale

Die Mikrokanal-Architektur benutzt ein asynchrones Protokoll für die Steuerung und Datenübertragung zwischen Hauptspeicher, E/A-Einheiten und dem System-Mikroprozessor.



(MC)

Abbildung 3-3. Der Mikrokanal

Funktionen

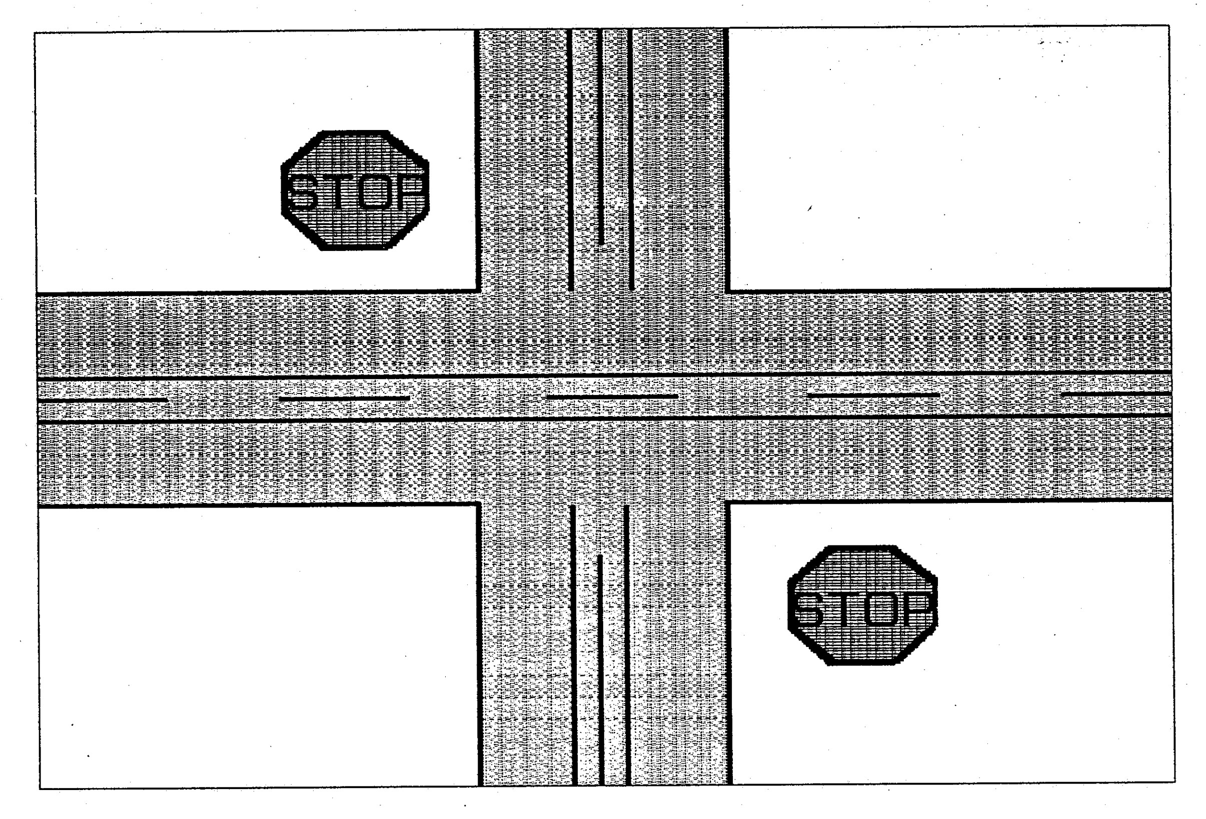
Hauptmerkmale

- Unterstützung einer zentral gesteuerten Arbitration (Schlichtungsstelle), die bis zu 15 MASTER bei der Nutzung des Mikrokanals überwacht.
  CACP Central Arbitration Control Point
- Level sensitive Interrupts mit Interrupt sharing auf allen Ebenen, d.h. zeitgleiche Unterbrechungen werden erkannt und verarbeitet.
- Asynchrone Übertragung = freie Geschwindigkeitswahl
- Fehlernachrichten und -behandlung (Recovery).
- E/A-Adressierung mit 32 Bit ermöglicht 8- oder 16-Bit oder 24-Bit oder 32-Bit E/A-Übertragung innerhalb 64 K Gesamtadressen.
- Hauptspeicher-Adressierung mit 32-Bit erlaubt 8- oder 16-Bit Speicher-Übertragung innerhalb 16 MB (oder 24- oder 32-Bit bis 4 GB) Gesamtadressraum.
- Architektur erlaubt I/O-Geschwindigkeiten bis zu 160 MB/sec(!).

3.4 BUS Architektur des Personal Computer

Entworfen für Single-Tasking Umfeld

- Synchrone Prozessor-Kontrolle, BUS-Takt Teil des Prozessor-Taktes
- Flankengetriggerte bzw. Impuls sensitive Interrupts
- Architektur erlaubt I/O-Geschwindigkeiten bis zu max. 33 MB/sec.



(BU)

Abbildung 3-4. Der PC BUS

Programmierbare Optionsauswahl (POS)

Hauptmerkmale:

- Keine HW-Schalter und -Brücken zu setzen auf Systemplatine und Adapterkarten.
- ADF Datei mit Konfigurationsdaten für jeden Adapter
- Eindeutige Adapter-ID, mit ADF-Dateinamen über- einstimmend
- POS-Identifikation durch Beschreibungsdatei auf der Referenz-Diskette @XXXX.ADF
- Konfigurationsdaten werden im CMOS gespeichert
- Test und automatische Rekonfiguration beim Einschalt-Selbsttest (POST)
- Verhindern des gleichzeitigen Zugriffs auf Adapterkarten mit gleichem ID
- Identifizieren jeder Karte in jedem Slot
- Pro Adaptertyp muß eine System-Konfigurationsdatei (.ADF, Adapter Definition File) und weitere Dateien für Diagnose-Funktionen (.DGS) auf einer 3.5″ Diskette vorhanden sein.
  Diese Dateien müssen auch von externen Adapterherstellern zur Verfügung gestellt werden.
- Bei Anwahl muß die Karten-ID gesendet werden (nur 0er Bit's werden interpretiert, bei 8 Bit sind damit 39.202 Kombinationen möglich)

3.5 System Konfiguration

Während der Systemkonfiguration werden folgende Schritte vom System-Konfigurationsprogramm durchgeführt. Das Programm ist auf der Setup Diskette des jeweiligen Modells enthalten und läuft menügesteuert ab.

- Lesen Adapter ID Nr.
- Auswählen **.ADF** Datei
- Zuteilen der Systemresourcen
- Erstellen der CMOS Eintragung für jeden Slot
- Kollidierende Adapter stillegen (SLEEP-Funktion)
- Prüfwert für CMOS errechnen und abspeichern

Beim Einschalttest (POST) führt die System Startup Logik folgende Schritte durch:

- Prüfen CMOS-RAM Inhalte
- Prüfen ID Nr. in jedem belegten Slot
- Laden Register der entsprechenden Adapter
- In Ordnung - System wird initialisiert
- Fehler - Fehlernachricht 165, Konfigurationslauf wiederholen

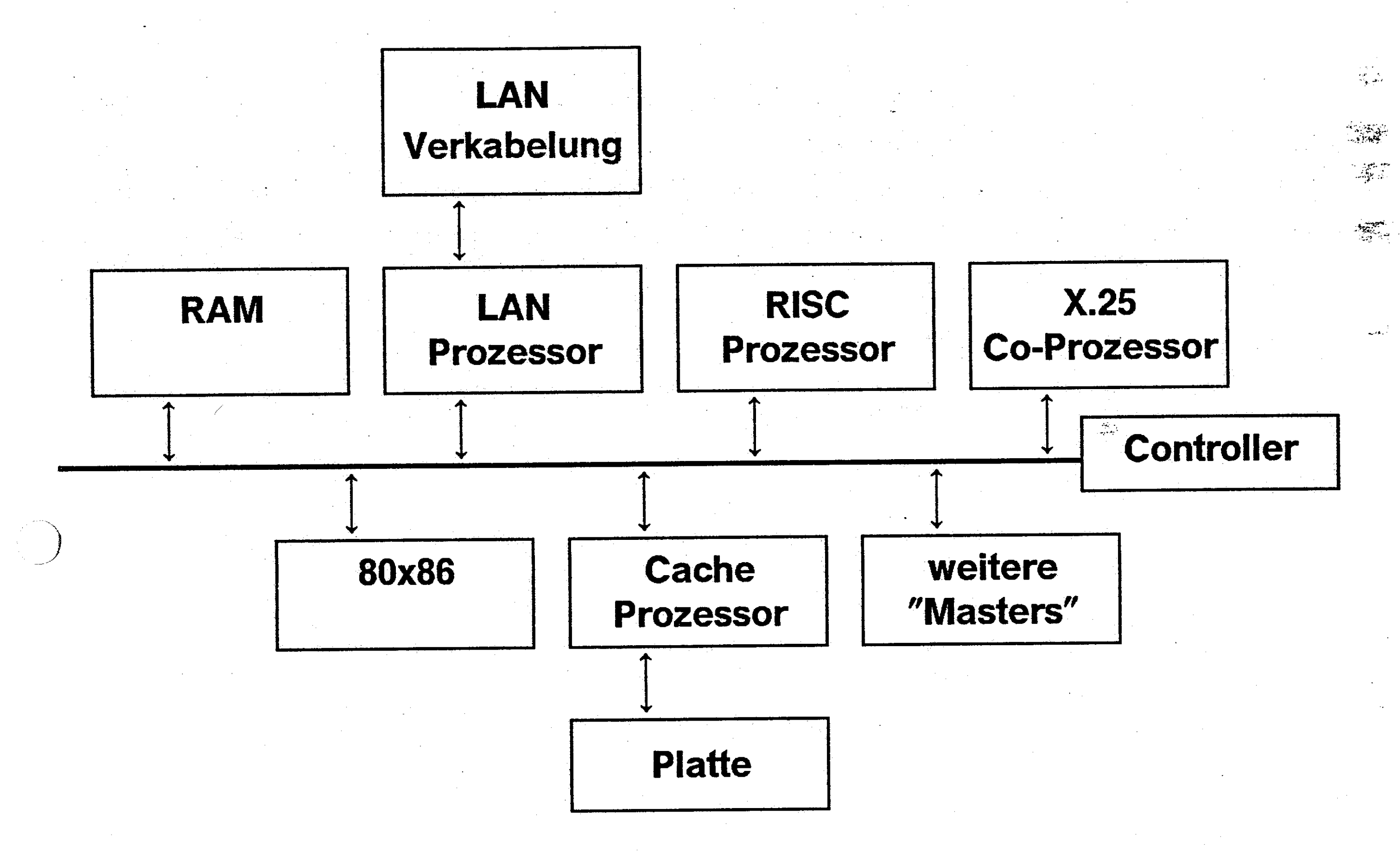
BUS Zeitüberschreitung

Wenn ein BURST Modus Arbiter die Kontrolle über den Kanal nicht abgeben würde, fände keine Speicherauffrischung mehr statt. Sogenannte „Soft"-Speicherfehler wären die Folge. Diese Einheit wird als „arroganter" Arbiter bezeichnet. Um dies zu verhindern, ist die Kanal-Zeitüberschreitung an den CACP angegliedert.

- max. 7,5 Mikrosekunden Kanalnutzung nach Aktivieren von -PREEMPT
  - dadurch garantiert mindestens 7,5 Mikrosekunden Kanalbenutzung (nur bei aktiven BURST)
- Speicherauffrischung findet alle 15,6 Mikrosekunden statt, Zeitgebergesteuert (Benutzt -PREEMPT)
  - hat höchste Priorität

Durch beide Funktionen wird gewährleistet, daß **kein** Arbiter länger als 7,5 Mikrosekunden nach Aktivieren von -PREEMPT den Mikrokanal blockieren kann.

MCA bietet Schnittstelle für vielfältige Komponenten:



Intelligente Subsysteme

- DASD Subsysteme
- LAN Management
- Kommunikations-Konzentratoren
- Prozessor-Upgrades
- Graphische Subsysteme
- Fehlertolerante Karten/Systeme
- Subsysteme für Encryption/Decryption

Der Mikrokanal ist notwendige und sinnvolle Voraussetzung für Wachstum und kontinuierlichen Leistungszuwachs

3.6 Hardware als Einflußfaktor

Letztendlich entscheidet immer das Anwendungsprofil auf Ihrem Rechner,

- arbeiten Sie hauptsächlich z.B. mit Grafikanwendungen, hilft Ihnen eine schnelle Grafikkarte mit entsprechend großem eigenen Speicher
- machen Ihre Anwendungen viele Plattenzugriffe, hilft Ihnen eine schnelle Festplatte
- Anwendungen im Bereich CAD/CAM werden durch einen schnellen Prozessor unterstützt
- „fahren" Sie OS/2 mit nur einer Anwendung, genügt vielleicht ein relativ kleiner Hauptspeicher, im Multitasking ist ein größerer Hauptspeicher von Vorteil

Bei Mischbetrieb sind es evtl. mehrere der aufgezählten Punkte, die Ihnen hilfreich sind.

Bedenken Sie auch, daß sich im Multitaskingbetrieb von OS/2 z.B. ein größerer Hauptspeicher wahrscheinlich positiver auswirkt, als die höhere Taktrate eines Prozessors.

CAD = *Computer Aided Design*

CAM = *Computer Aided Manufacturing*

Kapitel 4. OS/2 Version 2, Architektur OS/2

Ziele dieses Kapitels

- Dieses Kapitel zeigt einige der Architekturgrundlagen von OS/2
- Sie verstehen die grundlegenden Funktionen des Betriebssystems

Referenzen

- Red Book IBM OS/2 Version 2.0, Volume 1: Control Program, GG24-3730
- Red Book IBM OS/2 Version 2.0, Volume 3: Presentation Manager, GG24-3732

Inhalt dieses Kapitels

4.1 Architektur Betriebssysteme 4-3
4.1.1 Schichtenmodell 4-3
4.1.2 Privileged Levels 4-6
4.1.3 Speichermodell 4-8
4.1.4 Gemeinsamer Speicher, ... 4-11
4.1.5 Virtueller Speicher 4-14
4.1.6 Single- und Multitasking 4-17
4.1.7 Steuerung von 16- und 32-bit-Anwendungen . . . 4-22
4.2 Startvorgang von Hardware und Betriebssystem . . . 4-23

4.1 Architektur Betriebssysteme

Betriebssysteme sind nach verschiedenen Kriterien aufgebaut, erfüllen aber in jedem Falle, die Voraussetzung Verbindungen zu schaffen zwischen:

- der Hardware einerseits
- der Software, also den Anwendungen andererseits.

4.1.1 Schichtenmodell

Den Aufbau eines Betriebssystems kann man sich sehr gut als ein Schichtenmodell vorstellen.

1. Hier gehen wir auf das Modell des Betriebssytems DOS ein. Die Schichten bestehen aus:

   - Hardware, also Ein- und Ausgabegeräte
   - Gerätetreiber und das BIOS/ROM

     Im engeren Sinne ist das BIOS (*Basic Input/Output System*) bereits ein Betriebssystem. Es enthält die grundlegenden Funktionen die benötigt werden, um die Hardware mit der Software zu verbinden. Das BIOS ist im Gegensatz zum eigentlichen Betriebssystem DOS abhängig von der Hardware. Ganz wichtig aber, das BIOS ist, obwohl funktionell Software, der Hardware zuzuordnen und es ist fest im ROM (*Read Only Memory*) „eingebrannt".
   - Die nächste Schicht ist das DOS selbst. Es interpretiert die Anwendungsanforderungen und gibt diese entsprechend weiter.
   - Als oberste Schicht in unserem Modell, liegt die Anwendung auf dem Stapel.

   Ein Anwendungsprogrammierer kann unter DOS, durchaus am Betriebssystem vorbei an die Hardware gehen. Das ist in einer *Single Tasking* Umgebung auch kein Problem, da ja nur die eine einzige Anwendung die ansprechbaren Elemente, Input/Output, Bildschirmpuffer oder auch den Arbeitsspeicher verwendet. In einer *Multitasking* Umgebung darf so etwas nicht existieren, da selbstverständlich nur EINER die Kontrolle über alles haben kann. Sonst würde zwangsläufig nicht alles seinen richtigen Weg gehen, es würden sozusagen „krankhafte" Abläufe auftreten. Daher werden solcherart geschriebene Anwendungen als *ill behaved programs* bezeichnet.

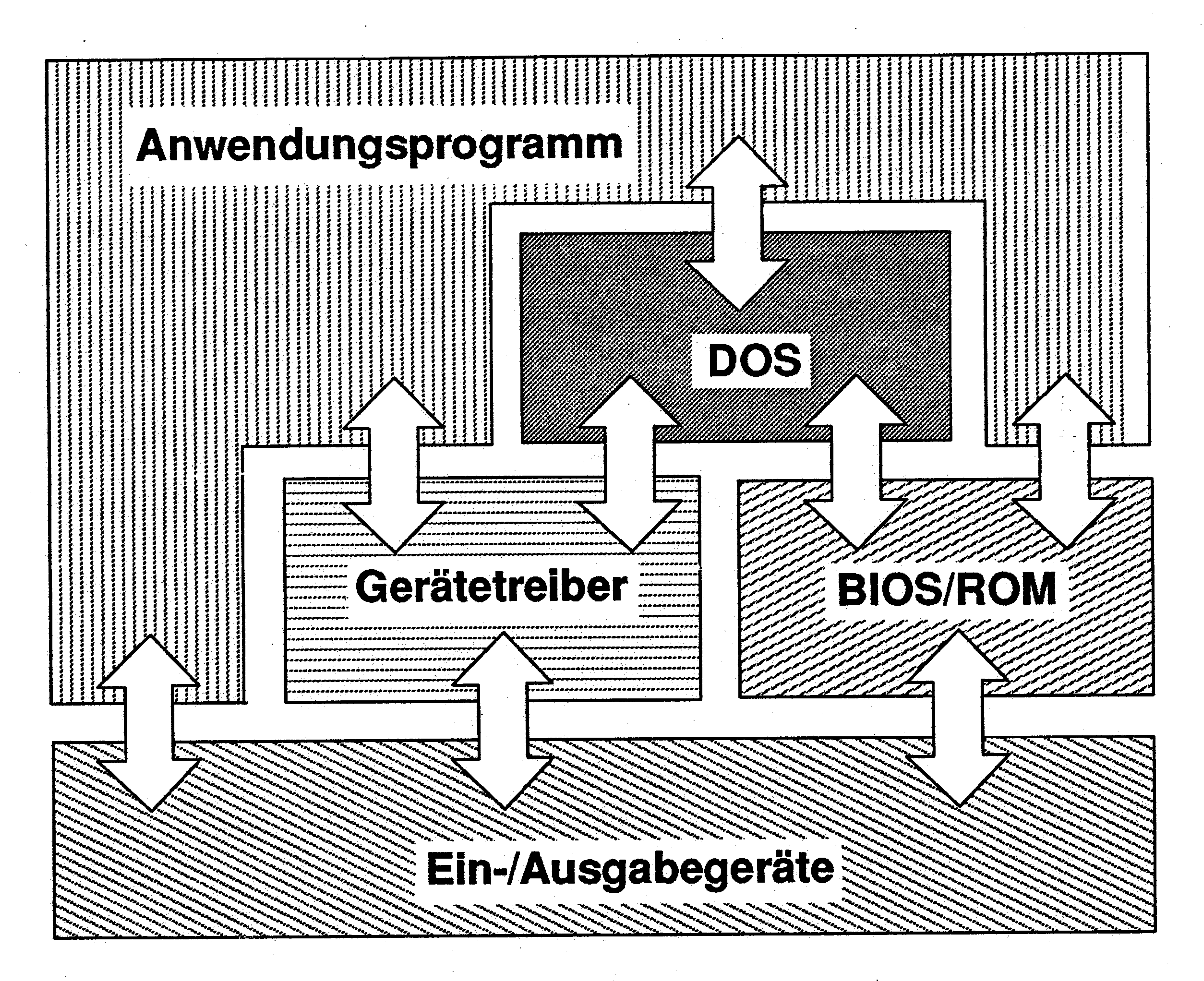


Abbildung 4-1. Beziehungen Software-Hardware unter DOS (J8HG9910)

2. Das Schichtenmodell für die Multitasking-Umgebung des OS/2.

   Auf den ersten Blick scheinen beide Schichtenmodelle annähernd gleich zu sein. Zwischen DOS und OS/2 gibt es auch im Schichtenmodell allerdings einen wesentlichen Unterschied.

   - Was bei DOS erlaubt ist, funktioniert aus soeben genannten Gründen bei OS/2 nicht, oder zumindest nur eingeschränkt. Einer Anwendung wird in keinem Fall erlaubt, am Betriebssystem vorbei an die Hardware zu gehen.

Es gibt allerdings eine Möglichkeit, einer Anwendung einen direkteren Input/Output-Zugriff zu erlauben. Dies wird in unserem Modell durch den IOPL-Block dargestellt.

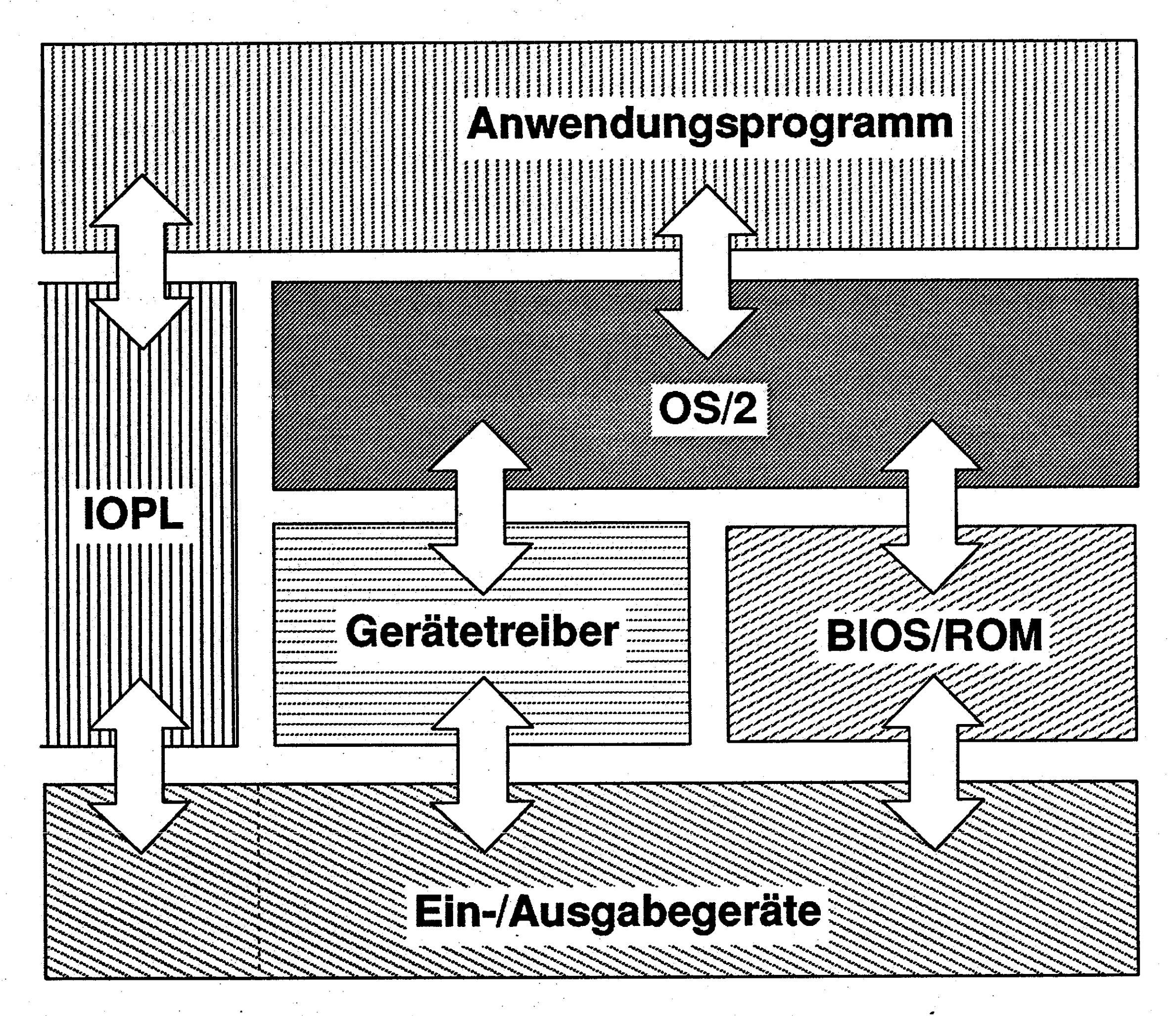


Abbildung 4-2. Beziehungen Hardware-Software unter OS/2 (J8HG9911)

Um zu verdeutlichen, auf welche Weise dieser IOPL-Zugriff erfolgt, behandeln wir das Kapitel Privileged Levels.

4.1.2 Privileged Levels

Der Prozessor kennt insgesamt 4 sogenannte *privileged levels*, also Schutzringe. Diesem Aufbau entspricht das Schutzkonzept des Betriebssystems.

- *privileged level* 0, Schutzring 0

  Auf dieser Ebene hat im wesentlichen nur der Kern des Betriebssystem etwas verloren. Andere Anforderungen dürfen auf Kernelfunktionen zugreifen aber keine Systemdateien anfassen.

- *privileged level* 1, Schutzring 1

  Diese Ebene wird von OS/2 nicht benutzt und ist für zukünftige Entwicklungen reserviert gehalten.

- *privileged level* 2 , Schutzring 2 Dies ist die Ebene, auf der betriebssystemnahe Programme laufen. Darüberhinaus laufen hier die Input/Output-Operationen ab, denen der Programmierer resp. der Benutzer-Service *Input Output Privileged Level* zugeordnet hat.

- *privileged level* 3, Schutzring 3 Die Ebene, in der die Anwendungen laufen.

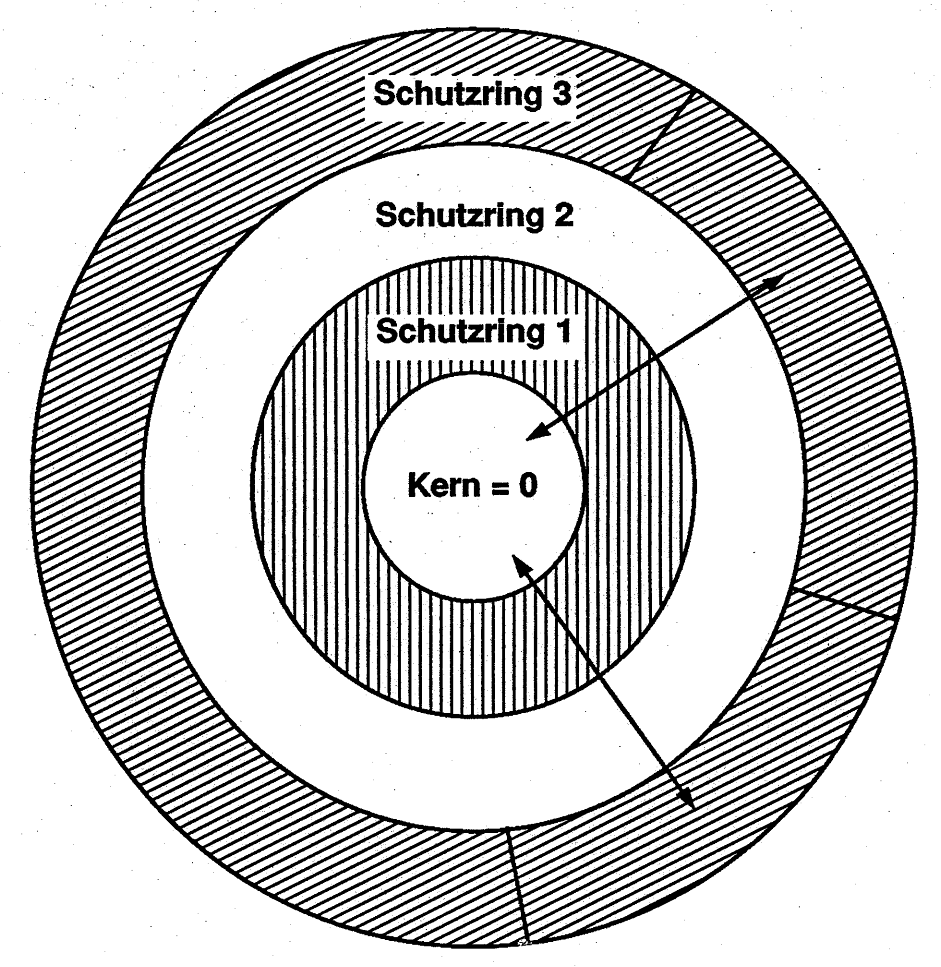


Abbildung 4-3. Schutzring-Konzept (J8HG9913)

4.1.3 Speichermodell

Basierend auf der Entwicklung von Hard- und Software gibt es 3 verschiedene Speichermodelle.

1. Konventioneller Speicher bis 1 MB

   Dieser wurde mit dem IBM PC entwickelt und war zu Beginn mit 16 kB ausgestattet. Er konnte auf der Hauptplatine bis 64 kB erweitert werden. Mittels Speichererweiterungskarten mit je 64 kB war es anfangs möglich, auf 256 kB zu erweitern. Zu diesem Zeitpunkt erschien eine systembedingte maximale Kapazität von 1 MB „gigantisch" zu sein.

   - Wenn wir auf dem 16-bit-Prozessor 8086 aufbauen, können hiermit $2^{16} = 65536$ bytes adressiert werden.
   - Durch Verwendung von 16 Segmenten mit jeweils 64 kB werden insgesamt 1 MB adressierbar.

Tabelle 4-1. Speicheradressierung

Segm.	Adresse	Beschreibung/kB
0	0000:0000 - 0000:FFFF	0 bis 64/Systemplatine
1	1000:0000 - 1000:FFFF	64 bis 128/Systemplatine
2	2000:0000 - 2000:FFFF	128 bis 192/Systemplatine
3	3000:0000 - 3000:FFFF	192 bis 256/Systemplatine
4	4000:0000 - 4000:FFFF	256 bis 320/Erw. Karten
5	5000:0000 - 5000:FFFF	320 bis 384/Erw. Karten
6	6000:0000 - 7000:FFFF	384 bis 448/Erw. Karten
7	7000:0000 - 7000:FFFF	448 bis 512/Erw. Karten
8	8000:0000 - 8000:FFFF	512 bis 576/Erw. Karten
9	9000:0000 - 9000:FFFF	576 bis 640/Erw. Karten
10	A000:0000 - A000:FFFF	zus. Bildschirmspeicher
11	B000:0000 - B000:FFFF	Bildschirmspeicher
12	C000:0000 - C000:FFFF	Festplattenadapter
13	D000:0000 - D000:FFFF	Verwendbar f. EMS
14	E000:0000 - E000:FFFF	Verwendbar f. EMS
15	F000:0000 - F000:FFFF	BIOS/Basic ROM

2. Speicherverwaltung unter dem Prozessor 80286

   Dieser Prozessor arbeitet mit 24 Adreßleitungen, und 16 Datenleitungen. Aufgrund der 24 Adreßleitungen adressiert er $2^{24} = 16.777.216$ bytes, also 16 MB. Er kennt bereits den *Protected Mode*.

3. Speicherverwaltung unter einem 32-bit Prozessor, 80386 oder höher

   Diese Prozessoren arbeiten mit 32-Adreßleitungen und können somit $2^{32} = 4.294.967.296$ bytes, also 4 GB physischen Hauptspeicher und 64 TB virtuellen Speicher pro Anwendung adressieren. Wie kann man sich die Adressierung vorstellen?

   - Das *directory* besteht aus den bits 22 bis 31 und spezifiziert die Adresse einer *page table* in einem *Page Directory*. Dieses enthält 1 k *Page Table-* Adressen.
   - Die *Page*-Adresse besteht aus den bits 12 bis 21 und spezifiziert den *Page Frame* innerhalb der vorher definierten *Page Table*. Eine *Page Table* enthält 1 k *Page Frame*-Adressen.
   - Das dritte Element, das *Offset* besteht aus den bits 0 bis 11 und definiert die Adresse innerhalb eines *Page Frames*
   - Eine *Page* enthält 4 kB.

   Die Multiplikation aus *Page Table* * *Page Frame* * *Page*-Größe ergibt:

   ```
   1 k * 1 k * 4 kB = 4 GB
   ```

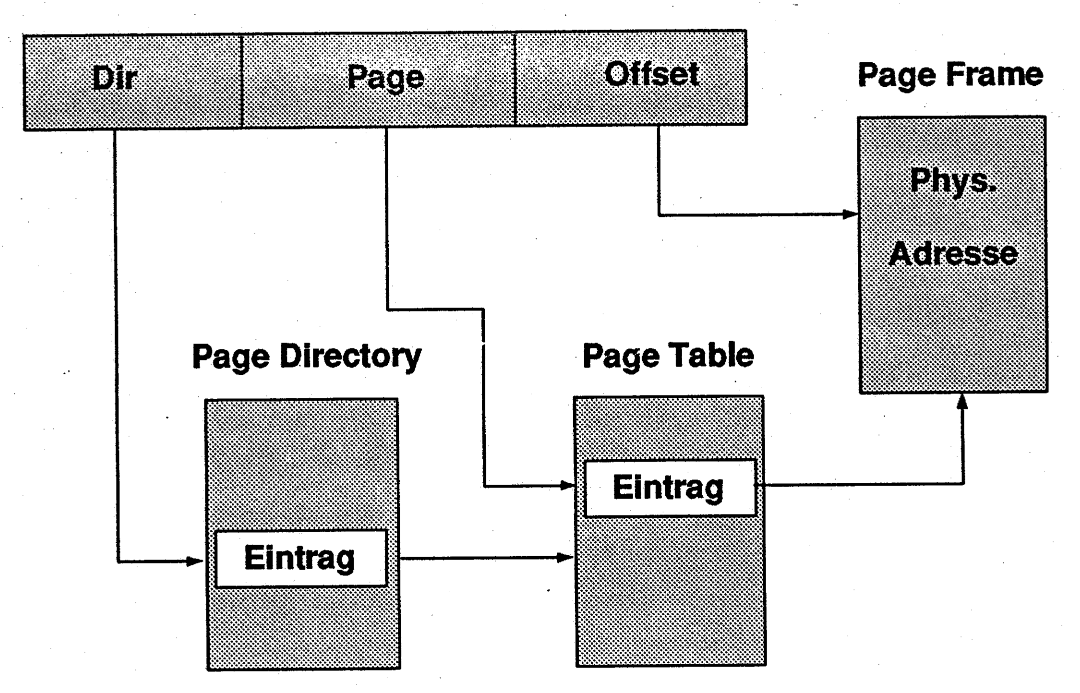


Abbildung 4-4. Adress-Umsetzung (J8HG9902)

4.1.4 Gemeinsamer Speicher,

Shared Memory

Die Speicherverwaltung von OS/2 wird vom *Memory Manager* vorgenommen. Dazu gehört als wichtiges Element das Konzept des Gemeinsamen Speichers, der den Datenaustausch zwischen Prozessen ermöglicht. (*interprocess communication*)

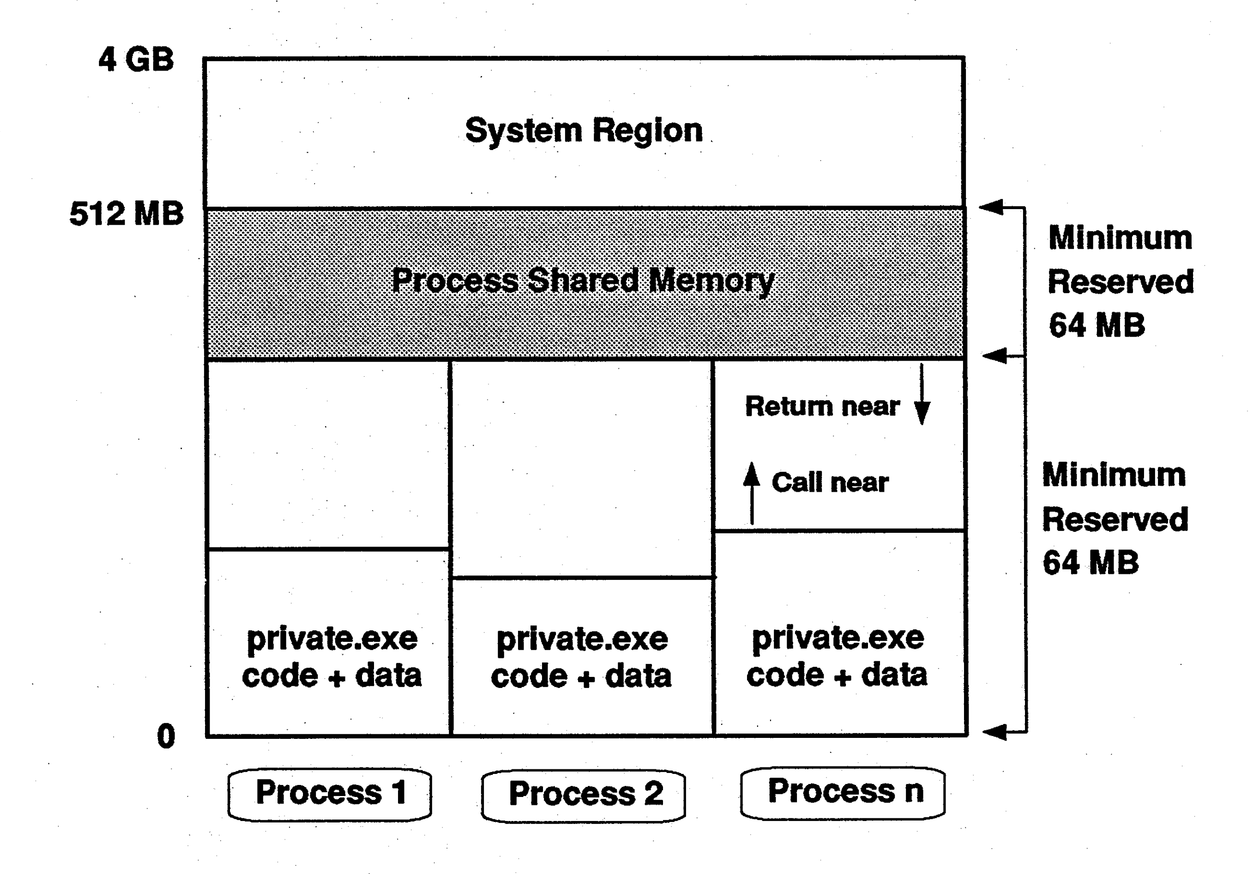


Abbildung 4-5. Shared Memory (J8HG9914)

Unter OS/2 gibt es zwei verschiedene Arten:

1. *named shared memory*
2. *anonymous shared memory*

Named shared memory

Der Zugriff zu dieser Art des gemeinsamen Speichers erfolgt über den Namen. Beide werden mittels einer virtuellen Speicheradresse im Adreßraum der verschiedenen Prozesse eingerichtet.

- Alle Prozesse, die den Namen des eingerichteten Bereiches kennen, können zugreifen.
- Das *shared memory* steht wird im OS/2 Filesystem registriert und steht dann zur Verfügung.
- Der Speicherbereich wird mit dem API - Aufruf DosAllocSharedMem eingerichtet.
- Mit dem API DosGetNamedSharedMem erfolgt der Zugriff.

Queues

Im Zusammenhang mit dem *named shared memory* werden *Qeues* als Steuerungselement eingesetzt.

- Mit dem API DosCreateQueue wird eine Queue eingerichtet.
- Das API DosOpenQueue ermöglicht den Zugang.
- DOSWriteQueue ist das API zum Schreiben.
- DosReadQueue erlaubt das Lesen.
- Wenn alle Prozesse, die auf das Segment zugegriffen haben, DosCloseQueue signalisiert haben, wird die Queue geschlossen.

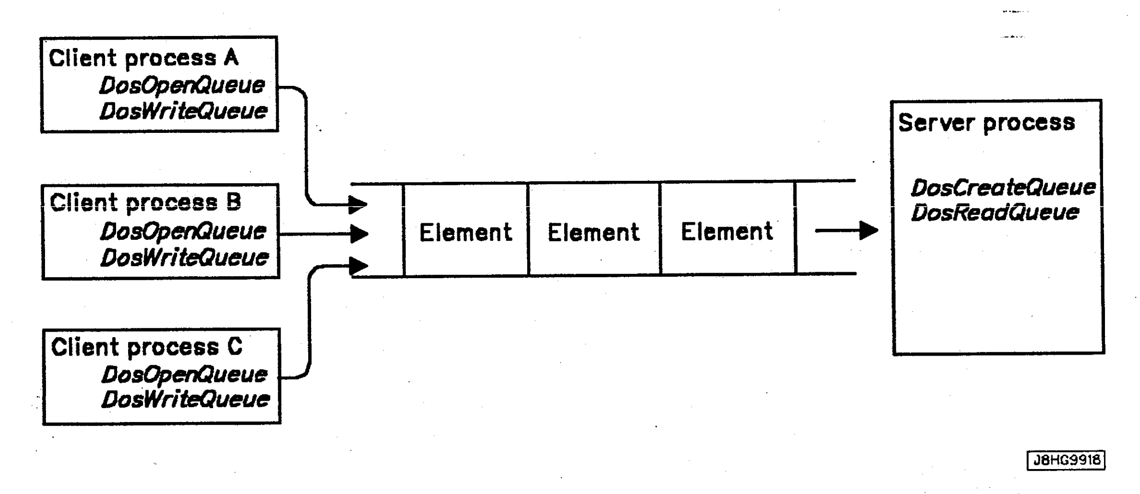


Abbildung 4-6. Funktion von Queues (J8HG9918)

Anonymous bzw. Give/Get Shared Memory

Dieser steht ausschließlich fest mit einander verbundenen Prozessen zur Verfügung.

- Die Verbindung wird mit Give/Get Shared Memory hergestellt.
- Das API DosGiveSharedMem stellt den Speicherbereich bereit.
- Das API DosGetSharedMem erteilt das Recht, auf Daten zuzugreifen.

Diese Art von *shared memory* ist zwangsläufig sicherer, als das *named shared memory*.

Pipes

Pipes sind Datenverbindungen zwischen 2 Prozessen. Auf die Daten kann nur sequentiell zugegriffen werden. Es gibt auch hier 2 verschiedene:

- *Anonymous Pipes*

  Sie werden auch *unnamed pipes* genannt, sie erlauben nur Abwärts-Datenaustausch.

- *Named Pipes*

  Sie erlauben den Datenaustausch auf gleicher Ebene und in Client- Server- Umgebung.

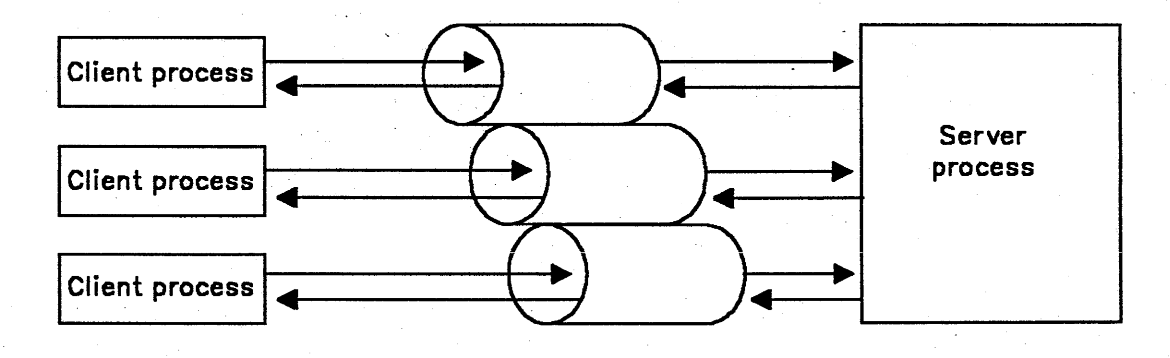


Abbildung 4-7. Named Pipes, 3 Kanäle und Duplex (J8HG9919)

4.1.5 Virtueller Speicher

Wie bereits behandelt, sind wir unter DOS auf einen konventionellen Speicher von 640 kB beschränkt und können maximal 1 MB adressieren. WINDOWS ermöglicht durch EMS, XMS, UMB die Bereiche zwischen 640 kB und 1 MB und darüber hinaus mit Einschränkungen zu nutzen. OS/2 stellt die Adressierung von bis zu 4 GB physischen Speicher zur Verfügung.

Abgesehen davon, daß 4 GB physischer Arbeitsspeicher heute noch nicht realisiert sind, stoßen Sie bei den meisten Maschinenkonfigurationen mehr oder schnell an die tatsächlich installierte Größe des Speichers.

Unter OS/2 stellt dieses aber kein Problem dar, da das Betriebssystem Virtuellen Speicher einrichtet. Dieser kann bis 64 TB groß sein. Sie werden diesem Prozeß auch als *memory overcommitment* wiederbegegnen.

Dafür stehen folgende Methoden zur Verfügung:

1. *Swapping*
2. *Paging*

Swapping

Unter OS/2 Versionen 1.x wird diese Methode genutzt.

- Das bedeutet, daß jeweils bis 64 kB große Datenblöcke bei Bedarf aus dem Arbeitsspeicher heraus auf die Festplatte gestellt werden. Das bedingt, bei größerem Auslagerungsbedarf die Nutzung von *huge memory* Zuordnung.
- Die Daten werden als Speicherabbild in die Datei SWAPPER.DAT gestellt.

Paging

Beim *Paging* werden wesentlich kleinere Datensätze ausgelagert. Es handelt sich um die Methode, die OS/2 ab Version 2.0 eingesetzt wird.

- Standardmäßig jeweils 4 kB.
- Das *Paging* ermöglicht, die Auslagerung gezielter vorzunehmen.

- Eine Reorganisation des Speichers ist nicht erforderlich.
- Obwohl es sich hier um funktionelles *Paging* handelt, heißt die Auslagerungsdatei nach wie vor SWAPPER.DAT.
- Die SWAPPER.DAT wird gewartet, d.h. sie wächst nach Bedarf und wird auch wieder kleiner.
- Die virtuellen Adressen stehen im *process virtual address space*.
- Virtuelle Adressen werden von der *Memory Management Unit* mittels *address translation* bei Bedarf in physische Adressen umgewandelt.
- Der erforderliche Schutz wird dadurch sichergestellt, daß Zugriff zu dem *virtual address space* nur durch einen *thread* ermöglicht wird, der im Kernel-Mode läuft.
- Ein Prozeß kann weder auf den *virtual address space* eines anderen Prozesses noch auf den des Systems zugreifen. Diese Regelung wird als *Memory Protection* bezeichnet.

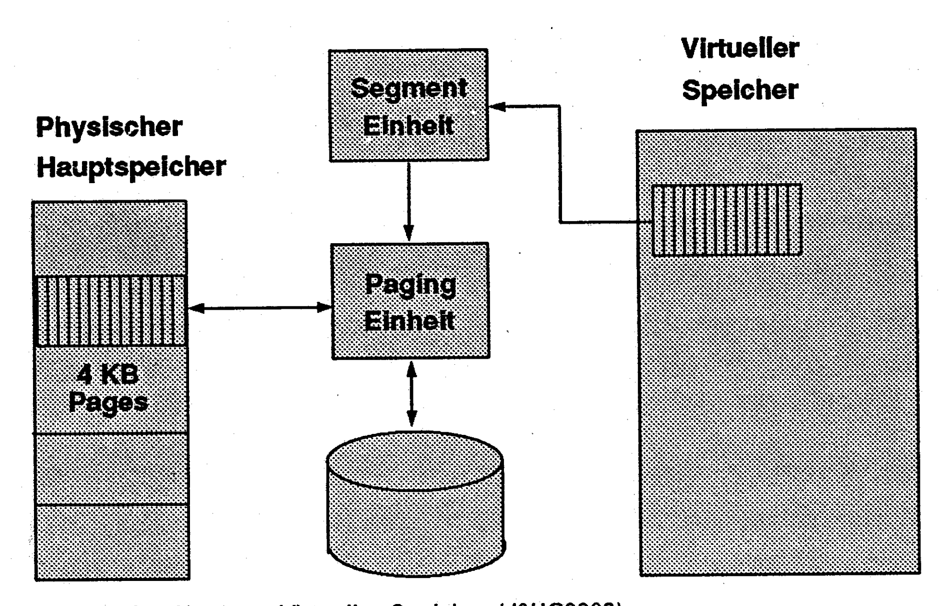


Abbildung 4-8. Paging-Konzept, Virtueller Speicher (J8HG9903)

4.1.6 Single- und Multitasking

In einer heutigen Betriebssystem-Umgebung gibt es grundsätzlich 2 verschiedene Arten von „Task"-Bearbeitung durch das Betriebssystem:

1. Singletasking
2. Multitasking

Singletasking

Zu Beginn der Personalcomputer-Zeit war es kaum denkbar, daß auf dieser Ebene, mehrere Anwendungen zur gleichen Zeit würden laufen sollen. Dafür gibt es schon seit relativ langer Zeit die HOST-Betriebssysteme wie VM, MVS, die mehrere Anwendungen gleichzeitig zur Verfügung stellen.

Singletasking wird durch mehrere Charakteristika gekennzeichnet.

- Die Anwendung wird gestartet und verfügt vom Anfang bis zum Abschluß über alle Resourcen.
- Daher kann keine zweite Anwendung gleichzeitig aktiv sein, wenn von der Ausnahme der *TSR* abgesehen wird.
- Sofern der Anwendungsprogrammierer sich sehr gut im Betriebssystem auskennt, kann er ohne Nachteile am Betriebssystem vorbei auf die Hardware zugreifen.
- Dadurch kann eine solche Anwendung durchaus schneller ablaufen, als wenn sie auf das Betriebssystem zugreift.
- Der Anwendungsprogrammierer muß sich während der Entwicklung Gedanken über die ihm zur Verfügung stehenden Resourcen machen.

Multitasking

Hervorgerufen durch die verschiedenen Entwicklungen bei Hard- und Software reicht *Singletasking* an den meisten Arbeitsplätzen nicht mehr aus. So gibt es seit einigen Jahren auch auf dem Sektor der Personal Computer *Multitasking*-Lösungen:

1. Kooperatives Multitasking
2. Preemptives Multitasking

Es ist wichtig, zu wissen, daß bei einem EIN-Prozessor-Konzept der Hardware, das *Multitasking* nicht auf Prozessor- (CPU-) Ebene stattfindet, sondern:

- auf allen anderen Ebenen im Rechner
- und sich für den Benutzer so darstellt

Kooperatives Multitasking

Es handelt sich hier um den Lösungsansatz, den WINDOWS benutzt. Zum Verständnis ist es wichtig, zu wissen, daß es sich bei WINDOWS nicht um ein Betriebssystem handelt. WINDOWS wurde entwickelt um:

- die Möglichkeiten, die die weiterentwickelte Hardware mittlerweile bot, zu nutzen.
- DOS als Betriebssystemsoftware weiter zu benutzen

Das kooperative Multitasking bedeutet, daß mehrere Anwendungen den Prozessorzugriff nacheinander erhalten, allerdings nur mir der Einschränkung, daß die einzelne Anwendung so geschrieben wurde, daß sie von sich aus den Prozessor wieder frei gibt.

Es setzt also Kooperation von Seiten des Anwendungsprogrammierers voraus!

Preemptives Multitasking

Ganz anders ist das preemptive Multitasking konzipiert. Hier übernimmt das Betriebssystem die komplette Steuerung. D.h. in unserem Fall erteilt OS/2 den Zugriff auf den Prozessor. Eine Anwendung kann nicht den Prozessor „festhalten", weil OS/2 prioritäten- und zeitscheibengesteuert eingreift. Allerdings arbeitet OS/2 nicht nur im Multitasking-, sondern auch im Multithreading-Betrieb im Gegensatz zum Betriebssystem UNIX, das im Singlethreading-Betrieb läuft. Bevor wir den Begriff Multithreading behandeln, was kann man sich unter einem *thread* vorstellen? Hier einige Begriffe, die den 3-stufig-hierarchischen Aufbau innerhalb einer Anwendung aufzeigen:

1. In einer **Sitzung** läuft Ihre Anwendung. Sie ist die Einheit, die sich die physischen Input/Output-Einheiten mit anderen teilt.

Damit sie Ihre Aufgaben auch komplett wahrnehmen kann, besitzt sie je

- einen **logischen** Bildschirmpuffer
- eine **logische** Tastatur
- eine **logische** Maus
- einen oder mehrere Prozesse, die der Sitzung zugeordnet sind, oder in ihr ablaufen.

Hintergrundsitzungen können also durchaus Ihre I/O-Operationen an die **logischen Einheiten** weitergeben. Erst bei Aufruf der Sitzung in den Vordergrund durch Übergabe des **FOKUS** werden diese an die **physische Einheit** übergeben.

2. Der Prozeß ist die Basis-Resource, die OS/2 der Sitzung zur Verfügung stellt, er entsteht beim Aufruf Ihrer Anwendung. Jeder Prozeß besitzt:

   - seinen eigenen Speicher
   - *threads*

3. Der *thread* stellt nun die kleinste ausführbare Einheit dar. Beim Eröffnen einer Sitzung entsteht erst einmal ein *thread*. Ein *thread* kann weitere „Unter-“ *threads* erzeugen, resp. aufrufen. D.h. nun nichts anderes, als das eine entsprechend geschriebene Anwendung mehrere *threads* gleichzeitig aktiv haben kann. Diese werden auch entsprechend mit Prozessorzeit bedient, laufen also im Multithreading-Betrieb ab. Ein *thread* kann 3 verschiedene Stati haben:

   - ready to run

     Dies ist der Status, den ein *thread* hat, der auf die Bedienung durch den Prozessor wartet.

   - running

     Lediglich ein einziger *thread* kann auf einer *uniprocessor* - Maschine diesen Status haben, da es sich um den handelt, der gerade vom Prozessor bedient wird.

   - blocked

     Ein *thread*, der darauf wartet, daß ihm Daten oder Ereignisse zugeliefert werden.

Prozesse

Ein Prozeß wird „geboren", wenn ein Programm geladen wird. Prozesse entsprechen innerhalb des Betriebssystems den Anwendungen. Sie teilen untereinander die zur Verfügung stehenden Resourcen, wie Speicher, Dateien, Semaphoren, *Queues* und *threads*.

Auch wenn ein Prozess in einer Hintergrundsitzung läuft, kann er doch Ein- und Ausgaben vornehmen. Diese werden in virtuelle Einheiten gestellt. In dem Moment, in dem der Fokus an die Hintergrundsitzung übergeben wird, übergeben die virtuellen Einheiten Ihre Daten an die physischen Einheiten.

Jeder Prozeß besitzt seine eigenen Elemente, wie:

- Speicher
- *threads*
- Dateisystem
- Strukturen zur Datenkommunikation zwischen den Prozessen.

Prozesse haben folgend genannte Charakteristika:

- Die *threads* eines Prozesses teilen sich die Resourcen.
- OS/2 unterstützt bis zu 4096 Threads gleichzeitig.
- Jeder Prozeß ist von anderen durch einen *process identifier* (PID)zu unterscheiden.
- Ein Prozeß erzeugt beim Programmaufruf einen *thread*, er kann allerdings mehrere *threads*, die parallel oder sequentiell ablaufen, erzeugen.

Prioritäten

Wie bereits behandelt, handhabt OS/2 die Steuerung der *threads* u.a. prioritätengesteuert.

Das Betriebssystem, genauer gesagt, der *scheduler* stellt nach Ablauf jeder Zeitscheibe (*timeslice*), die einem *thread* erteilt wurde, fest, welcher der anstehenden *threads* die höchste Priorität hat und bedient diesen als nächsten. Der „alte" *thread* wird, soweit er noch nicht bis zum Ende abgelaufen ist, wieder in den Kreislauf, den *round robin* gestellt. Er wird „preempted".

OS/2 kennt 4 Prioritätsklassen:

- *time critical priority class* Sie stellt die höchste Klasse dar.
- *server priority class* Die zweithöchste Klasse.
- *regular priority class* Die dritthöchste Klasse.
- *idle priority class* Die niedrigste Klasse.

Jede Prioritätsklasse hat 32 (von 0 bis 31) Prioritätsebenen und ihren eigenen *round robin* -Zyklus. Darüber hinaus gibt es Steuerungsmechanismen, die jederzeit eingreifen um Prioritäten zu vergeben bzw. zu verändern.

- *Priority boost*

  Regelt die Prioritäten innerhalb der *regular class*.

- *foreground boost*

  Vergibt der jeweiligen Vordergrund-Sitzung die höhere Priorität.

- *I/O boost*.

  Dieser Prozess vergibt einem *thread* die jeweils höchste Priorität innerhalb seiner Klasse.

- *starvation boost*

  Regelt die Prioritäten innnerhalb der *regular class* und gibt einem *blocked thread* eine Priorität, die unterhalb der *time critical class* liegt.

4.1.7 Steuerung von 16- und 32-bit-Anwendungen

Die vorliegende OS/2-Version sorgt selbständig für die korrekte Steuerung von 16- und 32-bit-Anwendungen. Dafür wird der *thunk layer* verwendet. Dieser Begriff ist nicht übersetzbar, da es sich um ein Kunstwort handelt. Wie kann man sich diesen vorstellen?

Es handelt sich um zwei miteinander verknüpfte Adreßtabellen, die dem jeweiligen 16- bzw. 32-bit-API ermöglichen, die richtige Adressierung vorzunehmen.

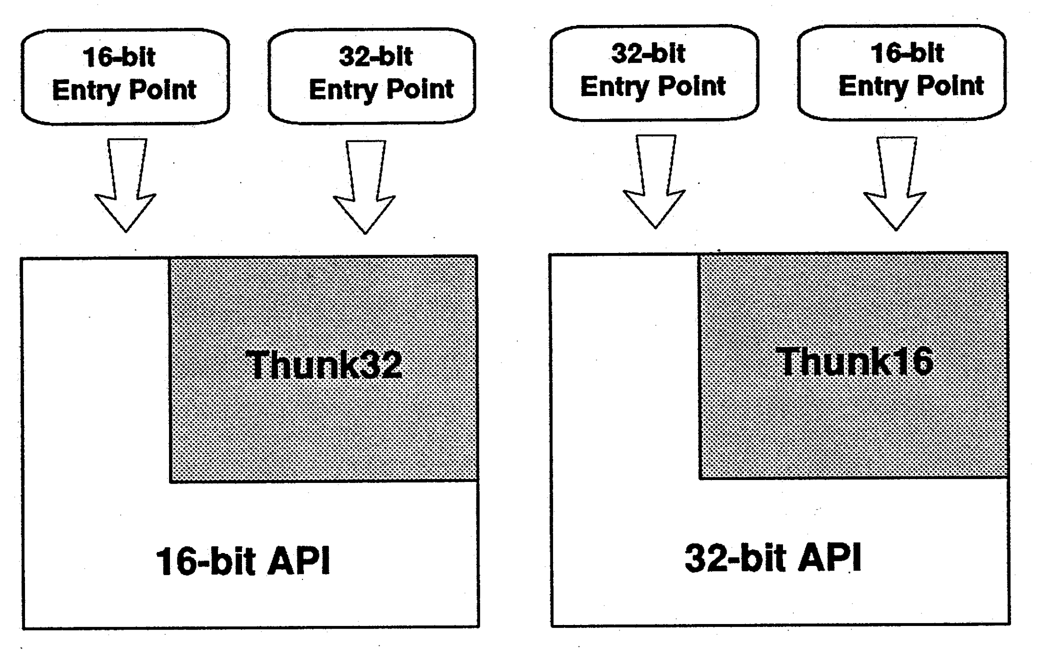


Abbildung 4-9. Thunk layer (J8HG9915)

4.2 Startvorgang von Hardware und Betriebssystem

1. POST, Einschalttest durch die System Startup Logik
2. Speicher-Rekonfiguration

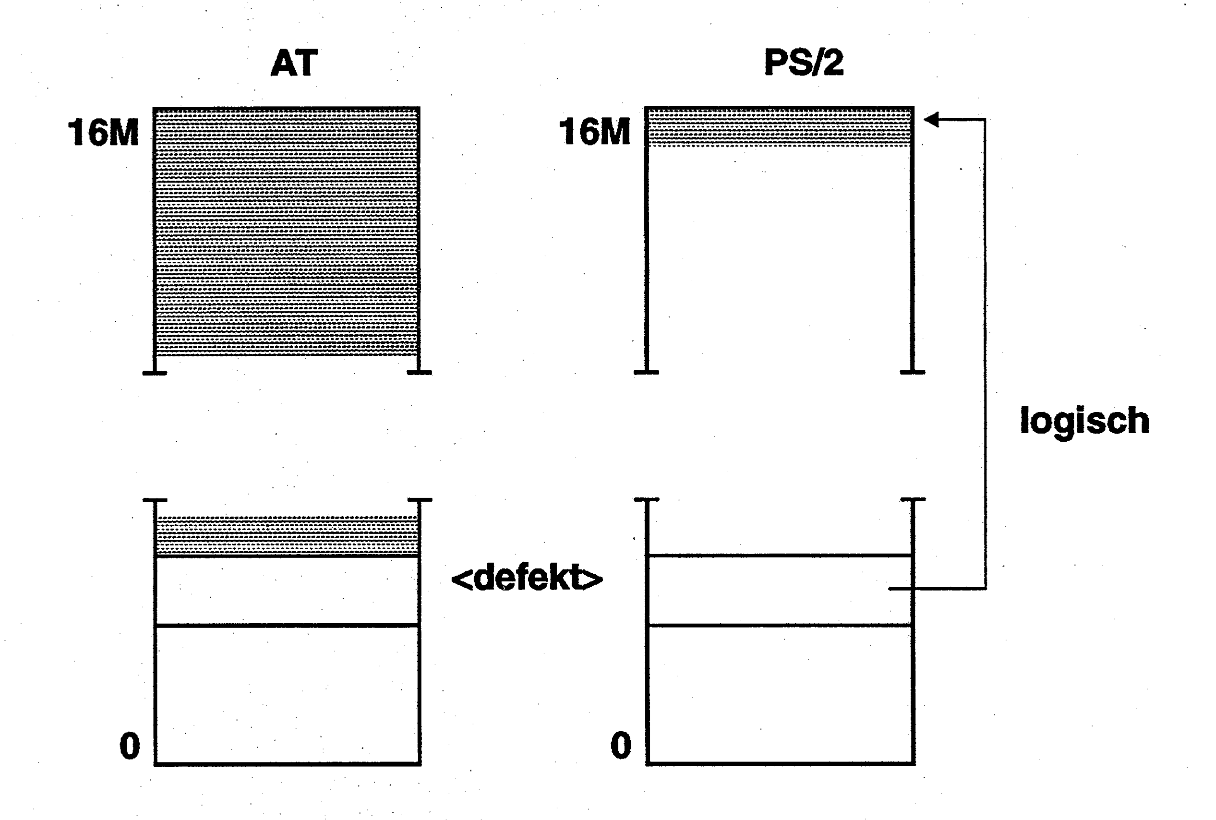


Abbildung 4-10. Speicher-Rekonfiguration (J8HG0350)

3. Prüfen der CMOS-Inhalte
   - **nichtflüchtiges** RAM
   - Stromversorgung durch 6 Volt Batterie
4. Inhalt des CMOS-RAM:
   - Systemuhr
   - Konfigurations-Informationen

- System Status Informationen
- geschütztes Passwort
  - 8 Byte + Prüfbyte
  - 3 Versuche

Cache-Speicher bestehen ebenfalls aus den sehr teuren nichtflüchtigen Halbleiterbausteinen, während der Hauptspeicher (RAM) aus weitaus billigeren Bausteinen besteht, deren Inhalt verloren gehen würde, wenn er nicht ständig „aufgefrischt" würde.

5. Prüfen der ID-Nr. in jedem Slot
6. Laden der Register der Adapter
7. Wenn i.O., Initialisierung des Systems
8. Wenn Fehler - Nachricht 165, Konfigurationslauf durchführen
9. Von welchem Laufwerk ist das Betriebssystem zu starten?
   - Diskette
   - Festplatte
10. Ggbf. Spur 0 der Festplatte Betriebssystemstart abfragen
11. Bootmanager?
12. OS2LDR
13. OS2KRNL
14. CONFIG.SYS
    - PMSHELL.EXE
    - CMD.EXE
    - HARDERR.EXE
15. SWAPPER.DAT

Kapitel 5. Drucken unter OS/2 Version 2

Ziele dieses Kapitels

Nach Durcharbeiten dieses Kapitels sollten die Teilnehmer:

1. Das Konzept des Druckens unter OS/2 Version 2 erklären können,
2. Beliebige Druckerobjekte anlegen und anpassen können,
3. Beliebige Druckerobjekte voll ausnutzen können.

Referenzen

- ITSC Red Book IBM OS/2 Version 2.0, Volume 5: Print Subsystem, GG24-3775

Inhalt dieses Kapitels

5.1 Allgemeines 5-3
5.2 Das Konzept der Druckausgabe 5-4
5.2.1 OS/2 - Schnittstelle 5-4
5.2.2 Übung: Installation eines Druckertreibers 5-13
5.2.3 Übung: Installieren eines Druckertreibers von einer Diskette aus 5-15
5.3 Verwenden von Trennseiten, separator pages 5-17
5.3.1 ÜBUNG: Erstellung einer Trennseite 5-19
5.4 Wege des Druckjobs 5-21
5.4.1 Übung: Drucken einer Datei einschließlich Wegverfolgung 5-23
5.5 Drucken im LAN 5-25
5.6 Drucken ohne Direktmanipulation 5-27
5.6.1 WIN-OS2 - Schnittstelle 5-27

5.1 Allgemeines

Das Konzept des Druckens wurde ganz auf eine Objektorientierte Basis gestellt. Es geht davon aus, daß für jede Direktmanipulation eine Quelle, d.h. eine zu druckende Datei, und ein **geeignetes** Ziel gebraucht werden. Diese Ziele sind sogenannte **Druckerobjekte**. Jedes Druckerobjekt ist eine selbständige Einheit, welche nicht unter der Steuerung eines zentralen Programmes läuft. Das erste Druckerobjekt wird bei der Installation des Betriebssystemes angelegt und mit Basiswerten eingerichtet.

Ein Druckerobjekt ist ein Symbol, das eine Anzahl von Festlegungen beinhaltet:

- Druckertreiber Objekt
- Schnittstellen Objekt, als physische Schnittstelle
  - Logische Schnittstelle
  - Physische Schnittstelle
- Druckjob Objekt
- Warteschlangentreiber Objekt
- Warteschlangen Objekt (*Spooler*)
- Anwendungs-spezifische Standard-Werte wie:
  - Papiergröße
  - Ausgabemaßstab (bei PostScript)
  - Format hoch - quer usw.

Das Drucken über einen LAN-Server ist durch spezielle LAN-Druckerobjekte so gestaltet, daß es für den Benutzer erkennbar ist, ob das Druckerobjekt ein LAN-Objekt ist und auch, ob dieses Druckerobjekt verfügbar ist.

Es werden alle weit verbreiteten Drucker und Plotter unterstützt. Die in der Version 2.0 vorhandenen fünf Drucker-Treiber-Disketten wurden durch Kompressions-Algorithmen bei Version 2.1 trotz einer Ausweitung der unterstützten Drucker auf lediglich zwei reduziert.

5.2 Das Konzept der Druckausgabe

Grundsätzlich gibt es zwei verschiedene Druck-Schnittstellen

- OS/2 - Schnittstelle in Form der Drucker Objekte
- WIN-OS2 - Schnittstelle in Form des Windows-Druckmanagers

Je nach individuellen Erfordernissen verwenden Sie entweder die eine oder die andere Schnittstelle, wobei in aller Regel die vom OS/2 zur Verfügung gestellte, die zu bevorzugende ist. Sie ist als Standard vom OS/2 bereits gesetzt.

Jede Anwendung soll dazu in der Lage sein, spezifische Druckausgaben zu machen. Das erfordert auf Seiten der Anwender ein Verständnis der Zusammenhänge.

5.2.1 OS/2 - Schnittstelle

Für jede Art der Ausgabe, die auf ein besonderes Papierformat oder auf einen besonderen Vordruck erfolgen soll, kann ein Druckerobjekt angelegt und angepaßt werden. Alle Druckerobjekte können auf einen einzelnen Drucker ausgeben. Deswegen hat jedes Druckerobjekt eine eigene Warteschlange, in der Druckausgaben festgehalten werden können bis der Drucker entsprechendes Papier hat. Es können auch verschiedene Druckerobjekte auf mehrere Drucker gehen.

Allen Druckerobjekten ist jedoch gleich, daß sie unter Steuerung des Benutzers verwaltet werden müssen, wenn mehrere verschiedene Dinge auf einen einzelnen Drucker geleitet werden.

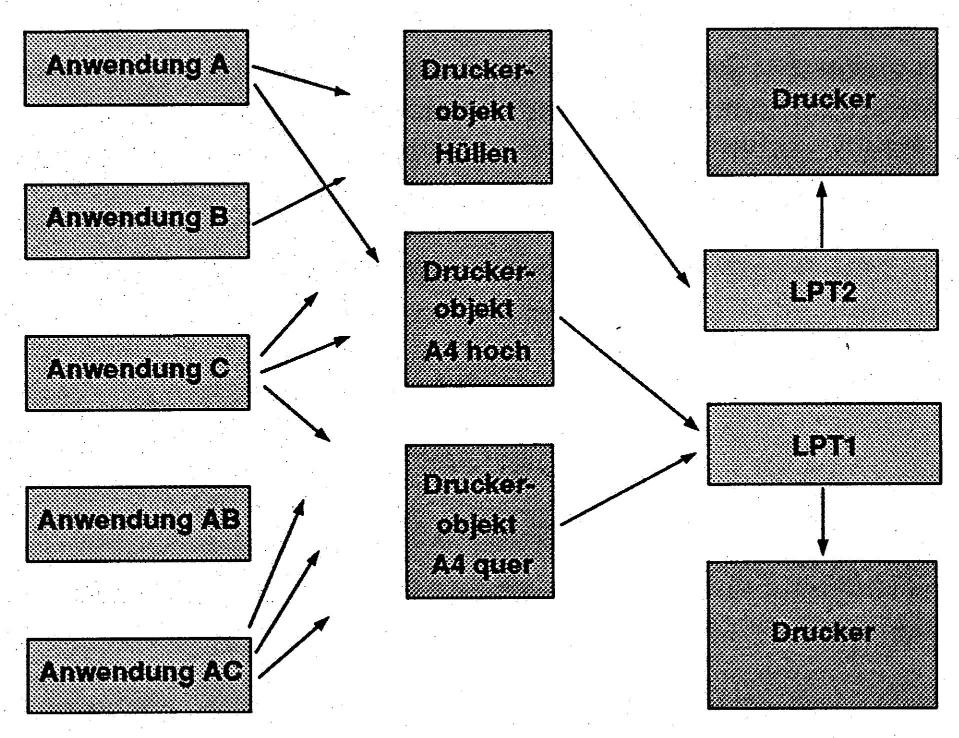


Abbildung 5-1. Viele Anwendungen - viele Druckerobjekte - mehrere Drucker (J81G0411)

Funktionen eines Druckerobjektes

Standardmäßig werden Druckerobjekte auf die Arbeitsoberfläche gelegt. Da die Symbole für mehrere Drucker gleich sein können, ist jedem Symbol ein **Objekttitel** als beschreibender Objekttitel angefügt. Das ist in Abb. 5-2 auf Seite 5-6 erkennbar. Für Ihre eigenen Fälle kann es aber durchaus sinnvoll sein, die **Druckerobjekte** in Ordner zu legen.

Kontext-Menü

Wie bei jedem Objekt, wird auch hier beim Anklicken mit Maus-Taste 2 ein Menü angezeigt, welches die momentan verfügbaren Auswahlmöglichkeiten anzeigt. Besondere Bedeutung kommt der Auswahl Öffnen - Einstellungen zu.

Die wichtigste Auswahl im normalen Betrieb ist das Anzeigen der noch zu druckenden Ausgaben.

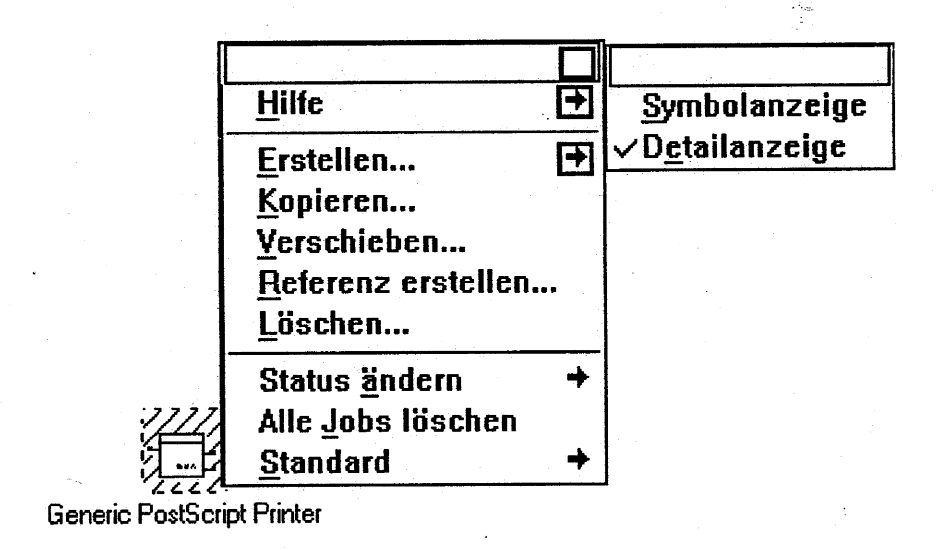


Abbildung 5-2. Kontext-Menü eines Druckerobjektes (J81G0401)

Abb. 5-3 zeigt die Notizbuchseite, die bei der Nachinstallation eines Drucker-Treibers benötigt wird. Von besonderem Interesse sind:

1. Installieren eines neuen Drucker-Treibers,
2. Festlegen der Schnittstelle (LPT1 usw.),
3. Festlegen des Treibers für die Ausgabe.

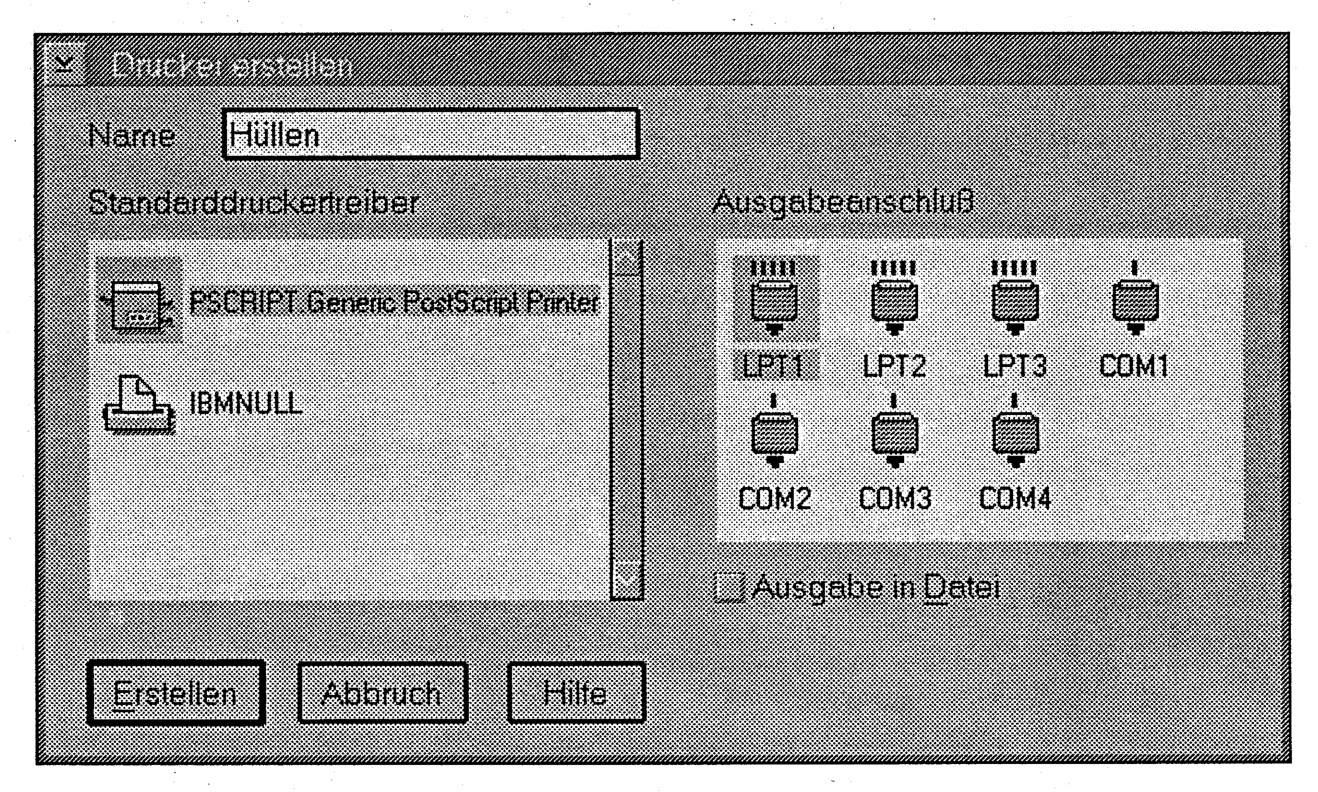


Abbildung 5-3. Festlegung des zu verwendenden Drucker-Treibers (J8HG0402)

Achtung

Obwohl es im Ordner Systemkonfiguration die Auswahl zum Installieren von Geräte-Treibern gibt, ist ein Drucker-Treiber **nur** über einen **angezeigten** Drucker-Treiber eines bereits bestehenden Druckerobjektes oder von einer Druckertreiberdiskette aus zu installieren.

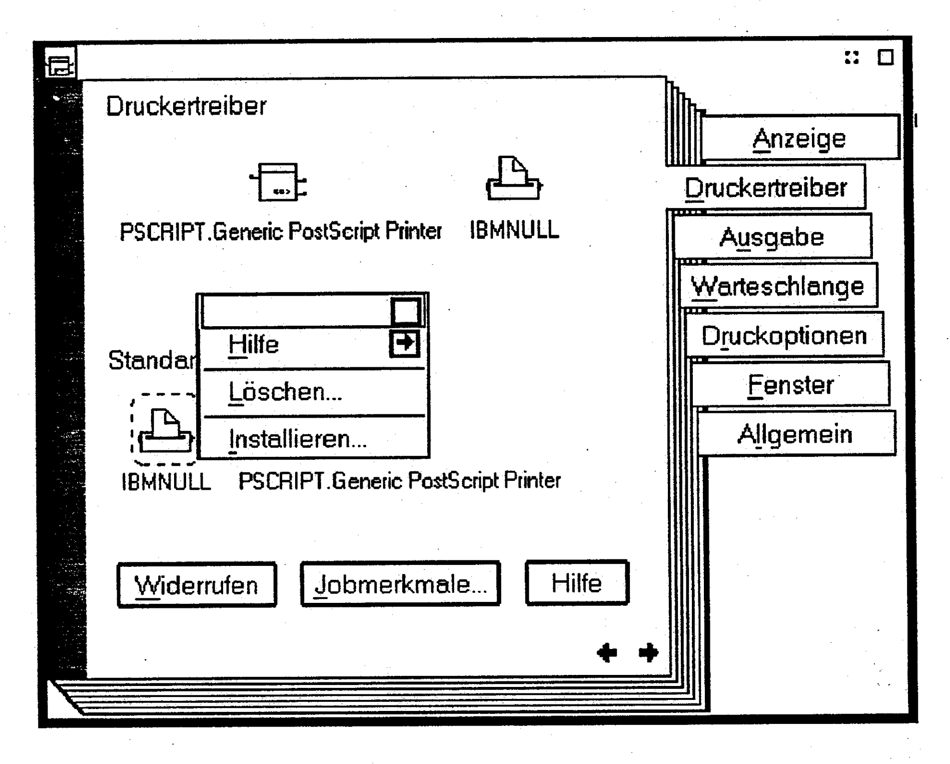


Abbildung 5-4. Kontext-Menü eines Drucker-Treibers (J8HG0403)

Status

Diese Auswahl gibt Auskunft über die Warteschlange für die Druckausgabe. Eine Warteschlange an einem Drucker-Server läßt sich nur von einem autorisierten LAN-Verwalter ändern.

Kopieren, Verschieben, Anlegen Referenzkopie

Soll eine Kopie des Objektes angelegt werden, welche später eventuell geändert werden soll, so wird über den Dialog das Ziel gefunden und festgelegt. Ist das Ziel sichtbar, so kann diese Funktion selbstverständlich auch mit der Maus direkt ausgeführt werden.

Normalerweise wird ein Drucker nur dann verschoben, wenn er **nur** in einem gesonderten Zusammenhang benutzt oder angeboten werden soll.

Das Anlegen eines Druckers in einem eigenen Arbeitsbereich kann sich auf das Kopieren einer Referenz beschränken, solange keine besonderen Festlegungen erforderlich sind.

Standard

Mit dieser Festlegung werden die DOS-Druckfunktionen wie der Befehl PRINT usw. in die richtige Ausgabe geleitet.

Einige Anwendungen erkennen nur den einen Drucker, welcher als Standard-Drucker festgelegt ist. Das ist ein Fehler von Seiten des Programmautors.

Aktivieren

Durch doppeltes Drücken der Maus-Taste 1 wird die Liste der noch nicht abgearbeiteten Druckausgaben angezeigt. Es ist möglich, eine Ausgabe im eigenen System festzuhalten, d.h. eine Druckausgabe kann unterbunden werden, solange dies gewünscht ist.

Installieren eines zusätzlichen Druckerobjektes

Um Druckausgaben in verschiedenen Formaten auszugeben, ist es sinnvoll, mehrere verschiedene Druckerobjekte zu definieren. Jedes Druckerobjekt soll dann nur für ein einziges Format benutzt werden. Die Anzahl der Druckerobjekte hängt direkt von den Anwendungen ab. Kriterien sind:

1. Hochformat
2. Querformat
3. Papiergrössen
4. Vordrucke.

Für Blankopapiere muß je Grösse und Ausrichtung ein Druckerobjekt angelegt werden. Das sollte auch Briefhüllen einschließen.

Je nach Betriebsumgebung kann es sinnvoll sein, für jeden Vordruck einen eigenen Drucker zu haben, um nicht immer wieder Papier wechseln zu müssen.

Es gibt mehrere Vorgehensweisen zur Installation zusätzlicher Druckerobjekte, welche zum gleichen Ergebnis führen:

1. Kopieren eines vorhandenen Druckerobjektes und anschließende Änderung,
2. Verwendung einer Schablone
3. Erstellen eines neuen Druckerobjektes, ausgehend von einem vorhandenen Druckerobjekt
4. Aktivieren eines Druckertreibers von einer Treiberdiskette aus.

In diesen Dialogen kann ein neuer Name für den neuen Drucker angegeben werden, während beim Kopieren zwar der **externe Name** geändert werden kann, nicht jedoch der **Druckername.** Das führt bei Anwendungen, welche nicht für OS/2 Version 2 geschrieben wurden, zu unklaren Druckerauswahlen.

Kopieren eines Druckerobjektes

Dieses Notizbuch wird nach der Auswahl Kopieren angezeigt. Das Notizbuch erlaubt das Plazieren der Kopie in einen beliebigen Ordner und auch das Zuordnen eines neuen Namens.

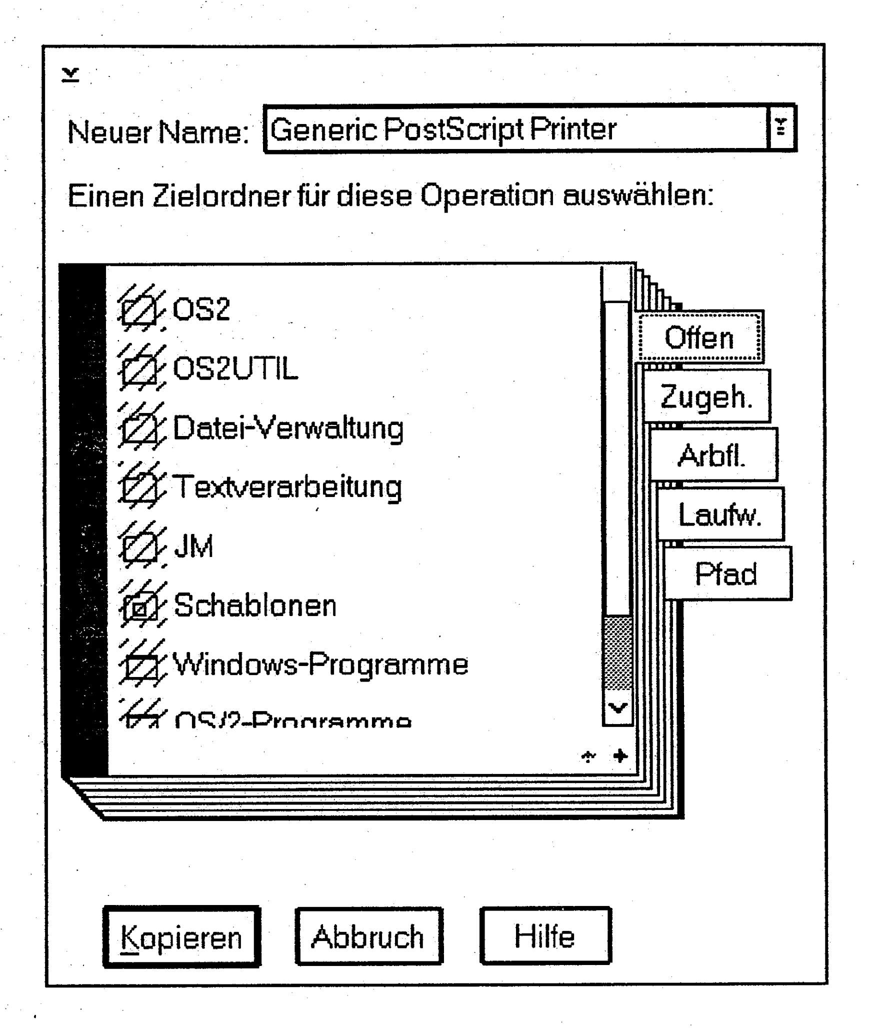


Abbildung 5-5. Notizbuch zum Kopieren eines Druckerobjektes (J8HG0405)

Beim Erstellen des neuen Druckerobjekt sind Treiber und Schnittstelle auszuwählen.

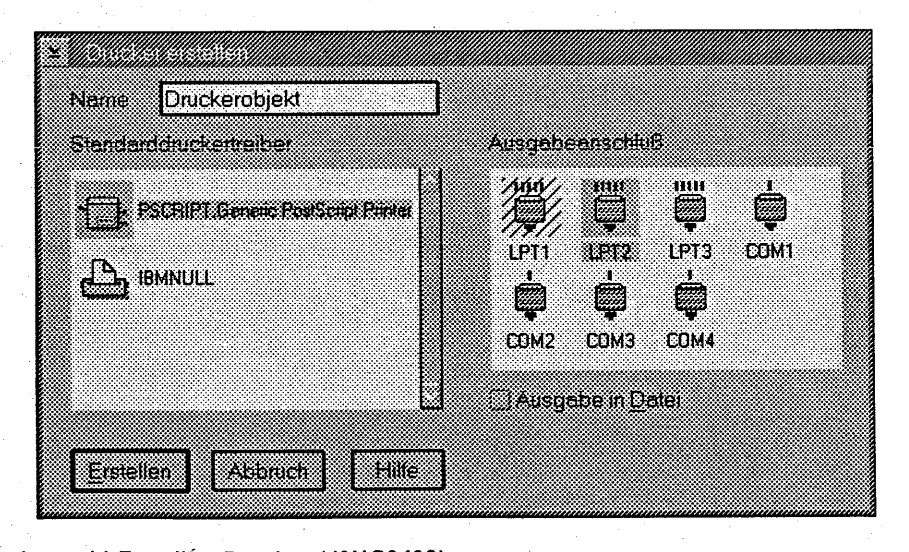


Abbildung 5-6. Auswahl Erstellen Drucker (J8HG0406)

Die Installation des LAN-Requester fügt neue Elemente in die Arbeitsoberfläche ein. Ein lokales Druckerobjekt sowie auch ein Druckerobjekt im LAN werden in dem Ordner Schablone zur Verfügung gestellt. Dadurch kann dann sehr einfach mit Direktmanipulation das gewünschte Druckerobjekt gleich an den richtigen Ort gebracht werden. Automatisch wird dann das erforderliche Notizbuch geöffnet.

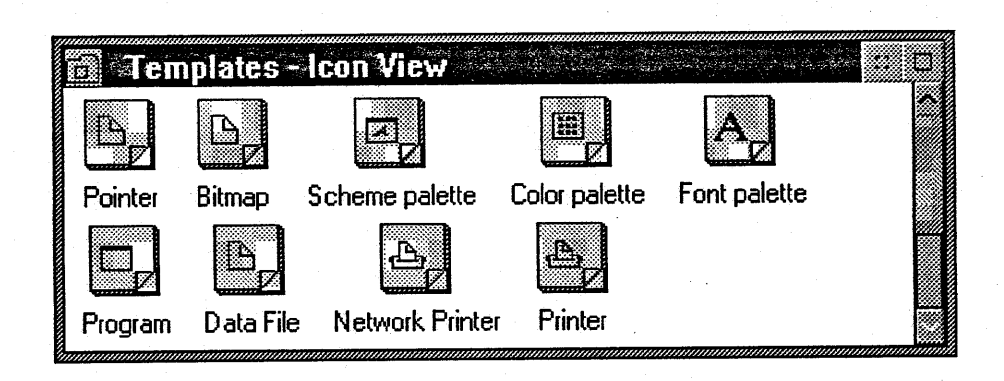


Abbildung 5-7. Ausschnitt aus dem Ordner Schablonen(J8HG0407)

5.2.2 Übung: Installation eines Druckertreibers

Ziel dieser Übung

1. In dieser Übung sollen die Kenntnisse aus „Installieren eines zusätzlichen Druckerobjektes“ vertieft werden.
2. Es soll ein Drucker-Treiber installiert werden

Vorbereitung

Die Übung sollte mit einer aufgeräumten Arbeitsoberfläche begonnen werden; d.h. alle Ordner geschlossen sein.

Wenn Sie sich über einen Ablauf nicht ganz im klaren sind, so versuchen Sie, die notwendige Information über Hilfe herauszubekommen.

Zur Vertiefung des Wissens ist es auch sinnvoll, die einzelnen Schritte und/oder Auswahlen in die vorgesehenen Zeilen einzutragen.

Durchführung

1. Öffnen Sie das Notizbuch Ihres Druckerobjektes,

   ____

2. Blättern Sie zur Seite Druckertreiber,

   ____

3. Öffnen Sie das Kontextmenü eines Drucker-Treibers,

   ____

4. Wählen Sie Installieren

   ____

   aus,
5. Sie brauchen nun die erste Diskette mit Drucker-Treibern des OS/2,

   ____

6. Wählen Sie bitte den Generic PostScript Printer aus oder einen Drucker, den Sie an Ihrem Arbeitsplatz haben,

   ____

7. Legen Sie fest, daß die Ausgabe in eine Datei erfolgen soll, da ja kein PostScript-Drucker zur Verfügung steht.

   ____

8. Geben Sie ein Textobjekt in das Druckerobjekt

   ____

9. Was geschieht?

   ____

   ____

5.2.3 Übung: Installieren eines Druckertreibers von einer Diskette aus

Ziel dieser Übung

1. In dieser Übung sollen die Kenntnisse aus „Installieren eines zusätzlichen Druckerobjektes“ vertieft werden.
2. Es soll ein weiteres Druckerobjekt unter Verwendung eines Drucker-Treibers auf einer Diskette installiert werden

Vorbereitung

Die Übung sollte mit einer aufgeräumten Arbeitsoberfläche begonnen werden; d.h. alle Ordner geschlossen sein.

Zur Vertiefung des Wissens ist es auch sinnvoll, die einzelnen Schritte und/oder Auswahlen in die vorgesehenen Zeilen einzutragen.

Durchführung

1. Aktivieren Sie die Druckertreiberdiskette 1 mittels des Objektes „Laufwerk A“.
2. Rufen Sie die Datei „PRDESC.LST“ auf. Sie erhalten somit die Liste der unterstützten Drucker.
3. Suchen Sie Ihren Drucker heraus, z.B. den

   „Seitendrucker IBM 3816 - 01S“

   und notieren Sie sich den Treiber:

   ____

4. Rufen Sie nunmehr die Datei „PRDRV.LST“ auf. Sie finden dort die laufende Nummer der Druckertreiberdiskette, die den benötigten Treiber enthält:

   ____

5. In unserem Fall benötigen Sie die Diskette Nummer 1, die bereits in Ihrem Laufwerk ist.
6. Suchen Sie den vorher herausgesuchten Treiber und aktivieren Sie ihn, wie immer mit Doppelklick der linken Maustaste.

7. Aus dem sich öffnenden Fenster ziehen Sie mit „drag and drop“ das korrekte Druckerobjekt auf Ihre Arbeitsoberfläche.

Als Ergebnis dieser Übung sollten Sie nun ein weiteres Druckerobjekt auf Ihrer Arbeitoberfläche haben.

5.3 Verwenden von Trennseiten, separator pages

In einer Multitasking Umgebung, insbesondere wenn in einem LAN mehrere Benutzer Druckjobs auf einen gemeinsamen Drucker legen, wird es erforder- lich die verschiedenen Druckjobs wieder sauber voneinander trennen zu können.

Hierfür gibt es die Möglichkeit Trennseiten zu definieren.

Mittels Trennseiten können Sie:

1. Druckjobs auf der eigenen Maschine (*stand alone*) voneinander trennen.
   - Sie können beliebig viele unterschiedliche Trennseiten definieren.
   - Dies setzt Sie in die Lage, gebündelte Druckjobs leicht wieder zuzuordnen.
2. Druckjobs von mehreren Maschinen/Benutzern voneinander trennen. Zusätzlich zu dem vorgesagten, ist hier folgendes zu beachten:
   - Im LAN hat nur der authorisierte Koordinator die Möglichkeit, Trennseiten zu aktivieren.
   - Über die Angaben auf der Trennseite, können die Druckergebnisse leicht auch von jedem Benutzer erkannt und dem Ersteller zugeleitet werden.

Hier die zu verwendende Syntax zur Erstellung von Trennseiten :

Code	Funktion
@	Kennzeichnet den Beginn der Datei
@L text	Druckt den hier definierten Text
@0	Beendet eine Zeile. Jede Druckzeile muß mit @L beginnen und mit @0 (NULL) enden.
@D	Druckt das Datum, an dem die Spooldatei erstellt wurde.
@T	Druckt die Zeit, zu der die Spooldatei erstellt wurde.
@N	Druckt den Namen des *submitter* und Benutzer oder Maschinen Identifikation.
@I	Druckt die laufende Nummer der Druckdatei innerhalb der Warteschlange
@H nn	Druckt druckerspezifische Steuerungsfolge. Wobei nn ein hexadezimaler Wert ist.
@W nn	Definiert die Druckbreite der Trennseite in Anzahl Zeichen. Möglich sind 1 bis 132.
@n	Läßt n Zeilen leer. Möglich sind 0 bis 9.
@E	Definiert das Ende einer Trennseite oder einer Trennseitendatei

5.3.1 ÜBUNG: Erstellung einer Trennseite

Ziel der Übung

Sie erstellen eine Druck-Trennseite und probieren diese aus.

Durchführung

Die Übung sollte mit einer aufgeräumten **Arbeitsoberfläche** begonnen werden. D. h. alle nicht benötigten Ordner sollten geschlossen sein.

1. Starten Sie den Systemeditor.
2. Als Steuerzeichen des Beginns einer Trennseiten-Datei geben Sie bitte ein: (Zur Erinnerung, @ können Sie mit mit der Tastenkombination: ALT + 64 erhalten)

   @

3. Als nächstes geben Sie bitte den Text ein, der auf der Trennseite erscheinen soll.

   @L Textverarbeitung, Erstellername:..........

4. Nun werden Leerzeilen definiert:

   @3

5. Hier legen Sie fest, daß auf der Trennseite das Datum der Erzeugung der Spooldatei gedruckt wird.

   @D

6. Nun noch einmal Leerzeilen.

   @3

7. Hier legen Sie fest, daß die Uhrzeit der Erzeugung der Spooldatei gedruckt wird.

   @T

8. Und nun geben Sie ein, daß dies daß Ende der Trennseite ist.

   @E

9. Speichern Sie die Datei ab und geben Sie den Pfad mit an:

   C:\TRENN

10. Nun öffnen Sie das Kontextmenü Ihres Druckerobjektes und geben auf der Seite Druckoptionen unter Datei für Trennseiten den Namen Ihrer Trennseiten-Datei und den Pfad an.
11. Schließen Sie das Notizbuch wieder.
12. Nun ziehen Sie z.B. die AUTOEXEC.BAT auf das Druckerobjekt.
13. Nun müßten Sie 2 Druckseiten erhalten. Nämlich die

    Trennseite

    AUTOEXEC.BAT

Hat alles geklappt?

5.4 Wege des Druckjobs

Wenn Sie einen Druck veranlassen, läuft er über:

- Warteschlange
- Druckerobjekt
- Treiber
- logische Schnittstelle
- physische Schnittstelle

auf den physischen Drucker und wird dann gedruckt. Anhand der folgenden Übung möchten wir Ihnen diesen Weg aufzeigen.

Druck-Konzept

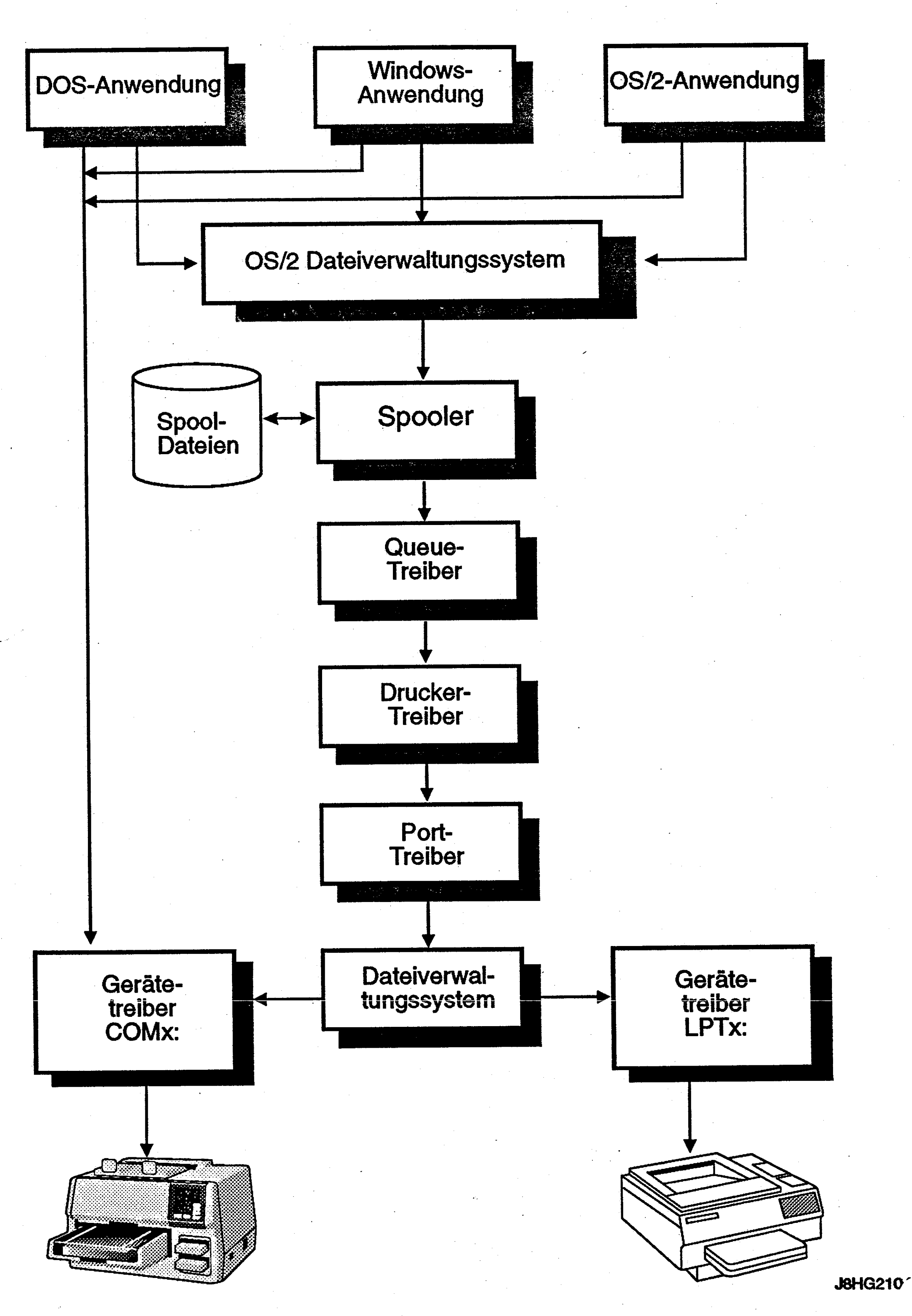


Abbildung 5-8. Wege eines Druckjobs (J8HG2104)

5.4.1 Übung: Drucken einer Datei einschließlich Wegverfolgung

Ziel dieser Übung

1. In dieser Übung sollen die Kenntnisse zum Thema „Drucken" vertieft werden.
2. Sie legen eine Datei auf ein Druckerobjekt und beobachten den Weg bis zum physischen Drucker.

Vorbereitung

Die Übung sollte mit einer aufgeräumten Arbeitsoberfläche begonnen werden; d.h. alle Ordner geschlossen sein.

Zur Vertiefung des Wissens ist es auch sinnvoll, die einzelnen Schritte und/oder Auswahlen in die vorgesehenen Zeilen einzutragen.

Durchführung

1. Öffnen Sie eines Ihrer Druckerobjekte und halten sie es an:
   - Systemmenü
   - Status ändern
   - Anhalten
2. Verkleinern Sie das Fenster und stellen Sie es sich an den Rand des Bildschirms.
3. Aktivieren Sie nun das Objekt „Laufwerke" und dann das Objekt Ihres aktuellen Betriebssystemlaufwerkes.
4. Dort finden Sie als Unterverzeichnis des Verzeichnisses SPOOL das Verzeichnis Ihres o.g. Druckerobjektes.
   - Wenn Sie sich nicht sicher sind, welches das richtige ist, gehen Sie in die Einstellungen des Druckerobjektes auf die Seite Anzeige. Dort finden Sie den physischen Namen, der mit dem Namen des Spool-Verzeichnisses identisch ist.
5. Stellen Sie sich die benötigten Fenster so auf den Bildschirm, daß Sie alle gleichzeitig sehen können.
6. Nun greifen Sie sich z.B. die Datei „CONFIG.SYS" und ziehen sie mit „drag and drop" auf Ihr (geöffnetes) Druckerobjekt.

7. Beobachten Sie was in den geöffneten Fenstern geschieht.

Sofern Sie einen Drucker physisch angeschlossen haben und dann das Druckerobjekt Freigeben wird Ihr Drucker die Arbeit erledigen.

5.5 Drucken im LAN

Die Benutzeroberfläche von OS/2 Version 2 wurde so gestaltet, daß bei der Installation des LAN-Requestor ein gesondertes Netzwerk-Symbol erscheint, welches die Ressourcen im LAN repräsentiert.

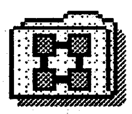


Abbildung 5-9. LAN - Ordner (J8HG0408)

Im Ordner NETWORK sind weitere Ordner. Zu Beginn ist nur der Ordner LS mit dem Ordner für alle LAN-Server vorhanden.

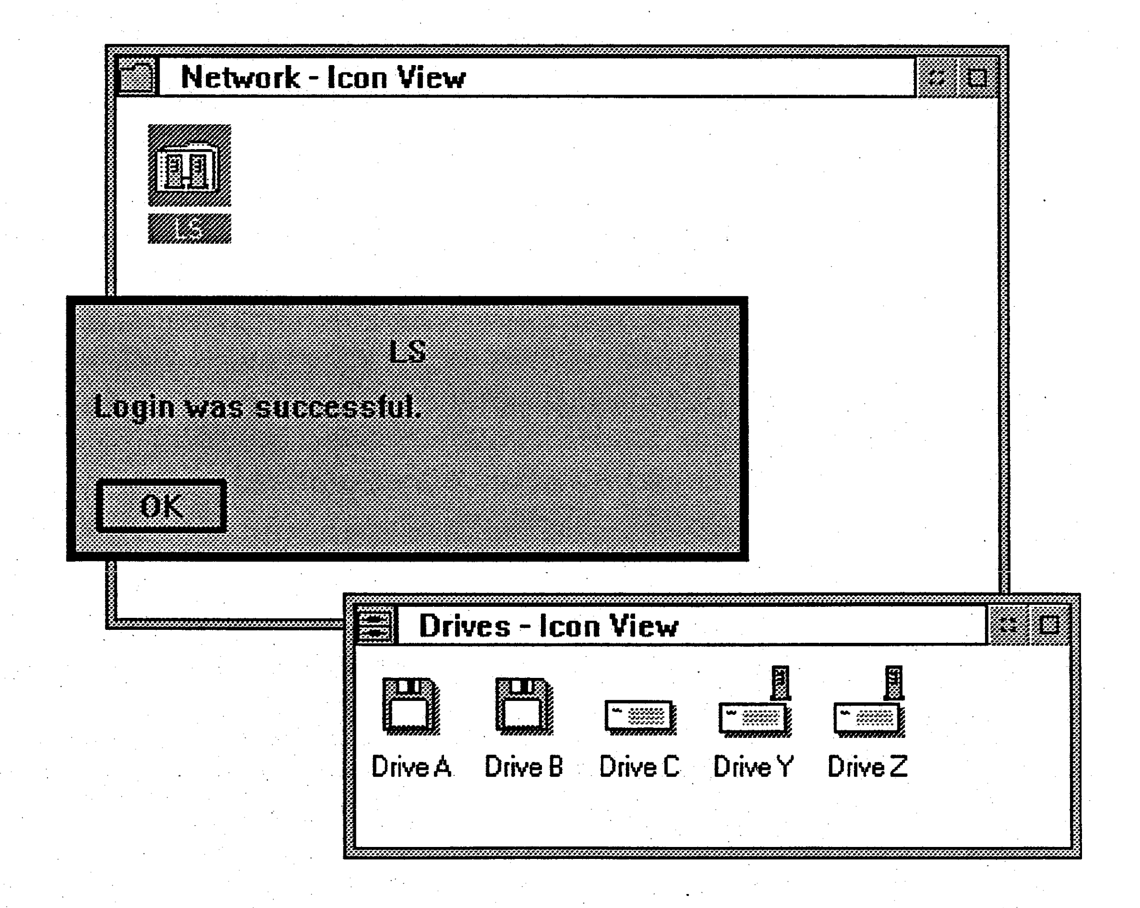


Abbildung 5-10. Netzlaufwerke (J8HG0409)

Wird dieses Symbol geöffnet, so erscheint ein Fenster, welches die vorhandenen LAN-Server zeigt. Nach Auswahl des gewünschten Server werden die Ressourcen dieses Servers gezeigt. Die Drucker werden mit den Namen ihrer Warteschlangen angegeben.

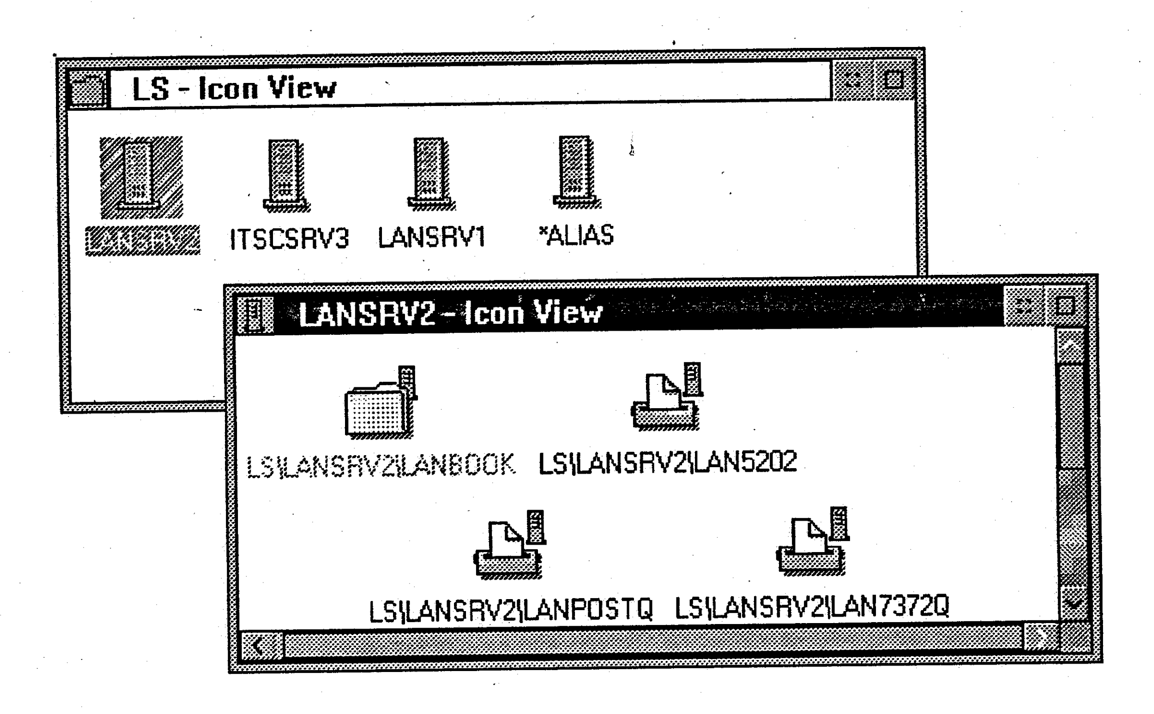


Abbildung 5-11. Netzdrucker (J8HG0410)

Soll einer dieser Drucker benutzt werden, so muss im eigenen, lokalen System ein entsprechender Drucker-Treiber vorhanden sein, um bei der Aufbereitung der Druckausgabe abgefragt werden zu können. Das Installieren eines Netzwerk-Druckers ist sehr einfach. Mit Drag-and-Drop wird ein Druckersymbol vom Netzwerk in einen lokalen Ordner gebracht. OS/2 fordert dann automatisch den Druckertreiber an.

Achtung

Die zu verwendenden Drucker sollten möglichst bei der ersten Installation mit definiert werden. Zumindest sollten die Druckertreiber im lokalen System vorhanden sein.

5.6 Drucken ohne Direktmanipulation

Nicht jeder Text kann einfach auf einem Druckerobjekt abgelegt und gedruckt werden. Nur wenn ein Text schon richtig formatiert ist, oder keiner Formatierung bedarf, kann er auch sofort gedruckt werden. Daher sind die meisten Anwendungen darauf angewiesen, eigene Formatierungen und eigene Druckausgaben zu machen. Das geschieht in traditioneller Art über eine Menüauswahl.

Es gibt außerdem eine Menge von Funktionen, welche nach den alten Regeln von DOS arbeiten. Es handelt sich dabei um:

- OS/2- und DOS-Befehle wie:
  - COPY xxx LPTn bzw. COMn
  - PRINT xxxx
  - Umleiten

    Dies setzt voraus, daß Sie mehr als eine Druckerschnittstelle belegt haben. Sie können wie gewohnt aus einer Befehlszeile heraus umleiten:

    `SPOOL >LPTn bzw. >COMn`

    aber auch objektorientiert:

    Gehen Sie in das Notizbuch des betreffenden Druckerobjektes auf die Seite Ausgabe und mit dem Mauszeiger auf den Port der dem Druckerobjekt zugeordnet ist und rufen Sie das Kontextmenü auf. (Zur Erinnerung: rechte Maustaste)

    Sofern Sie mehr als einen Port belegt haben, enthält das Kontextmenü den Punkt Umleitung.
- Ausgaben direkt zu einem Standard-Drucker
- Drucken von DOS aus

5.6.1 WIN-OS2 - Schnittstelle

Von der WIN-OS2 Seite aus gibt es grundsätzlich zwei Möglichkeiten der Druckwege:

1. Über den als aktiv definierten Windows-Drucker auf den OS/2-Spooler zu drucken.

Dies ist der empfohlene Weg, er stellt die größere Flexibilität dar und setzt Sie in die Lage, alle von den Anwendungen vorgesehenen Alternativen zu nutzen.

2. Den Druck-Manager zu benutzen.

   Wie zu Beginn des Kapitels bereits erwähnt, steht nach der Inaktivierung des SPOOLERS für WIN-OS2 - Sitzungen der Druck-Manager zur Drucksteuerung zur Verfügung. Wichtige Charakteristika des Druck-Managers:

   - es können mehrere Druck-Warteschlangen angelegt werden
     - diese können angehalten und freigegeben werden
     - zugeordnete Druckjobs können nach Markierung gelöscht werden.
   - aber immer nur eine davon pro Port kann aktiv sein
   - Über Optionen und Anzeigen können Sie das Verhalten und die Darstellung des logischen Druckers beeinflussen

Große Sorgfalt muß auf die Installation von logischen Druckern gelegt werden. D.h. Sie müssen sicherstellen, daß Sie die korrekten Treiber installieren. Dafür sind unter WIN-OS2 die Treiber zu installieren, die sich auf die gleichen Drucker unter OS/2 beziehen.

WINDOWS-Anwendungen sind mit stark unterschiedlichen Maßgaben geschrieben worden und verhalten sich dementsprechend:

- Zugriff auf definierten Standarddrucker
- Zugriff auf den erstmalig definierten Standarddrucker
- Möglichkeit aus allen installierten Druckern einen auszuwählen

Da bei der Vielzahl der Anwendungen keine allgemeingültigen Regeln aufzustellen sind, werden Sie für Ihre speziellen Fälle die richtigen Treiber und logischen Drucker- resp. Warteschlangen ausprobieren müssen.

Ziehen Sie auch die Dokumentation Ihrer Anwendungen zu Rate.

Kapitel 6. DOS und Windows unter OS/2 V2

Ziele dieses Kapitels

Dieses Kapitel macht Sie mit dem Einrichten und Benutzen der Virtuellen DOS-Maschine (VDM) sowie mit dem Benutzen von Windows-Programmen under OS/2 Version 2 vertraut.

Referenzen

- ITSC Red Book
  IBM OS/2 Version 2.0,
  Volume 2: DOS and Windows Environment, GG24-3775

Inhalt dieses Kapitels

6.1 Allgemeines 6-3
6.2 Migration von Programmen nach OS/2 PM 6-4
6.3 Virtuelle DOS-Maschinen 6-5
6.3.1 Einrichten der Virtuellen DOS-Maschinen 6-7
6.3.2 Einbinden von DOS-Programmen in die Arbeitsoberfläche 6-8
6.4 Speichermanagement unter DOS und WINDOWS 6-9
6.4.1 Upper Memory Blocks 6-10
6.4.2 High Memory Area 6-11
6.4.3 Expanded Memory Storage 6-12
6.4.4 Extended Memory Specification 6-13
6.4.5 DOS Protected Mode Interface 6-13
6.4.6 DOS-Einstellungen 6-14
6.5 Booten von virtuellen DOS-Maschinen 6-21
6.5.1 ÜBUNG: Virtuelles Booten eines DOS von Diskette 6-22
6.5.2 ÜBUNG: Virtuelles Booten eines DOS vom Image 6-23
6.6 Windows-Programme unter OS/2 Version 2 6-27
6.6.1 Einrichten der Windows-Umgebung 6-27
6.6.2 Datenaustausch zwischen OS/2-Programmen und Windows-Programmen 6-28
6.7 Installation von Windows-Programmen 6-31

6.1 Allgemeines

Um die Weiterverwendung aller DOS-Programme sowie aller Windows-Programme sicherzustellen und um diese außerdem in die Umgebung von OS/2 Version 2 zu integrieren, wurden die entsprechenden Umgebungen geschaffen. Diese unterscheiden sich von den Originalen in mehrerlei Hinsicht:

- Eine Virtuelle DOS-Maschine bietet mehr Hauptspeicher als DOS, normalerweise ca. 612 KB.
- DOS-Programme nehmen voll am Multi-Tasking teil .
- Bessere Sicherung gegen „Abstürze“ .

Die Programme können dem Benutzer auf verschiedene Arten zur Verfügung gestellt werden:

1. DOS-Programme können aus einer DOS-Befehlszeile aufgerufen werden,
2. DOS-Programme können in einer Gruppe an der Arbeitsoberfläche durch Symbole dargestellt sein,
3. Windows-Programme können in einer Gruppe an der Arbeitsoberfläche durch Symbole dargestellt sein,
4. Windows-Programme können als Symbole im Programm-Manager eines Fensters für Windows abgelegt sein,
5. In einer Sitzung Windows-OS/2-Gesamtbildschirm kann der „normale“ Windows-Programm-Manager benutzt werden.

6.2 Migration von Programmen nach OS/2 PM

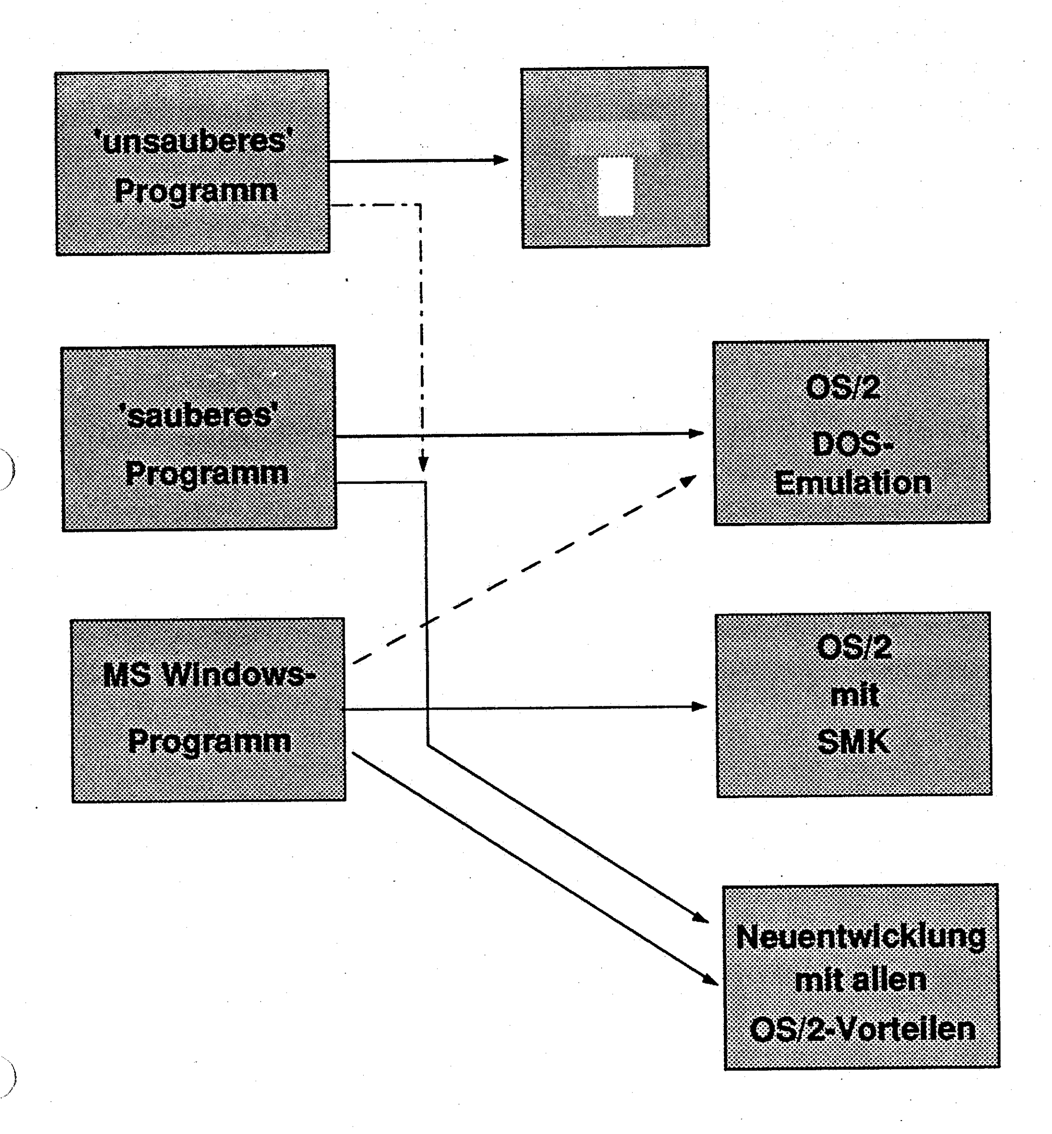


Abbildung 6-1. Migration von DOS nach OS/2 (J8HG1107)

6.3 Virtuelle DOS-Maschinen

Eine Virtuelle DOS-Maschine, kurz VDM genannt, ist im Grunde eine OS/2-Sitzung, in welcher DOS und ein 8086-Prozessor nachgebildet (emuliert) werden. Es werden hierzu, wie auch für andere Besonderheiten des OS/2, Einrichtungen des Microprozessors 80386 bzw 80486 benutzt. Die Emulation hat den Vorteil, daß mehrere DOS-Programme gleichzeitig arbeiten können und daß ein Fehler in einer VDM keine Nebenwirkungen auf andere Programme im System hat. Durch eine Einbindung in die Arbeitsoberfläche kann auch eine einfachere Benutzung eingerichtet werden.

Wenn eine Virtuelle DOS-Maschine aktiviert wird, so wird dafür von OS/2 ein gesonderter Speicherbereich angelegt. Dieser ist fest vergeben, gleich, ob ein Programm in diesem Bereich läuft oder nicht. Auch die Größe eines solchen Programmes spielt keine Rolle. Daher sind Virtuelle DOS-Maschinen sinnvollerweise nur dann zu öffnen, wenn sie benötigt werden oder der verfügbare Speicher keine Beschränkung darstellt.

Die Komplexität der Funktionen ist in Abb. 6-2 auf Seite 6-6 zu erkennen.

- Dem OS/2-Kern wird bei der Installation des Betriebssystemes (wenn angefordert) der *Enhanced DOS Session Kernel* angefügt.
- Wenn ein DOS-Programm gestartet wird, so stellt OS/2 Version 2 fest, daß es sich um ein solches handelt,
- OS/2 Version 2 verzweigt in die Emulations-Routinen und arbeitet dort wie DOS.
- Der Zugriff auf die HW ist nur dem OS/2 Version 2 gestattet, daher werden alle I/O-Funktionen über Virtuelle Gerätetreiber (*Virtual Device Drivers*) abgehandelt, welche die Ein- und Ausgaben dann über die OS/2-spezifischen Physischen Treiber (*Physical Device Drivers*)abwickeln.

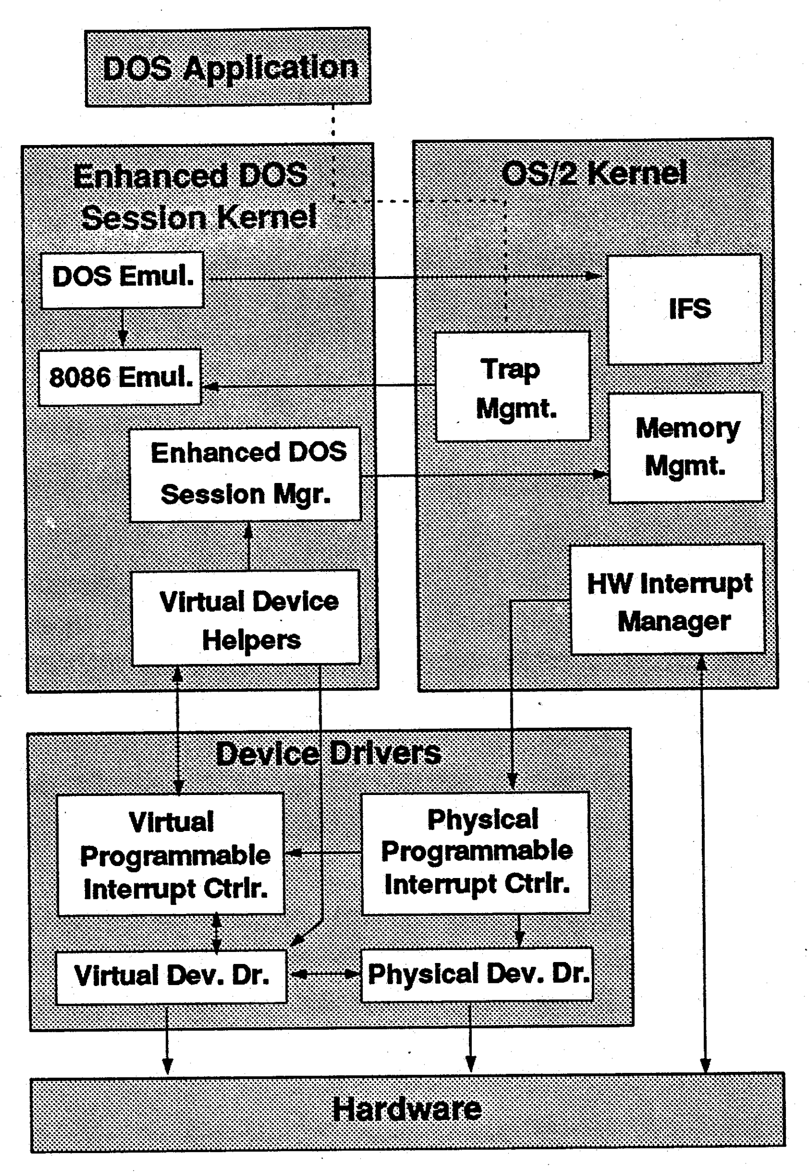


Abbildung 6-2. Konzept der Virtuellen DOS-Maschinen (J8HG0505)

6.3.1 Einrichten der Virtuellen DOS-Maschinen

Bei der Installation des OS/2 Version 2 wird angegeben, ob Virtuelle DOS-Maschinen eingerichtet werden sollen. Es werden Standardwerte verwendet, welche in fast allen Fällen *keiner* Anpassung bedürfen.

Soll dennoch eine Anpassung vorgenommen werden, so ist das in zwei Umgebungen möglich:

1. Nachinstallation
   - Ordner Systemkonfiguration
   - Installation anpassen
   - Wahl der DOS/WINDOWS-Unterstützung
2. Direkteingriff in die Umgebungsdefinition
   - Ändern der Datei CONFIG.SYS.
   - Autoexec.bat definieren
   - DOS-Objekt erstellen

Die Änderungen in der CONFIG.SYS werden in Kapitel 7, „Systemanpassung" auf Seite 7-1 besprochen. In Abb. 6-7 auf Seite 6-14 sind einige Parameter gezeigt, welche zur Anpassung verwendet werden können.

Viele Programme unter DOS und Windows benutzen alle möglichen Tricks, um mit dem vorhandenen Hauptspeicher und dessen Segmentierung klar zu kommen. Entsprechend sind die Einstellungen in einer VDM vorzunehmen, um den verschiedensten Anforderungen gerecht zu werden.

Ein DOS-Objekt wird nach folgendem Schema eingerichtet:

1. Ordner DOS-Programme anlegen (wenn nicht schon bei Installation mit vorhandenen DOS-Programmen automatisch geschehen)
2. Im Ordner Schablone das Symbol Programm anwählen und ein Objekt in den gewünschten Ordner abziehen,
3. Einstellungen für das Programm-Objekt öffnen,
4. Namen des DOS-Programmes oder der BAT-Datei eintragen und ggf. Parameter vorgeben,
5. Nach dem Schließen des Notizbuches kann das DOS-Programm durch Doppelklick aufgerufen werden.

6.3.2 Einbinden von DOS-Programmen in die Arbeitsoberfläche

Jedem DOS-Programm kann bei Bedarf ein Symbol auf der Arbeitsoberfläche zugeordnet werden. Dadurch wird der Zugriff zu diesen Programmen vereinfacht. Es kann sinnvoll sein, daß aus Sicherheitsgründen nur solche DOS-Programme verwendet werden, welche keinen Abbruch und damit Zugang zur DOS-Befehlszeile erlauben.

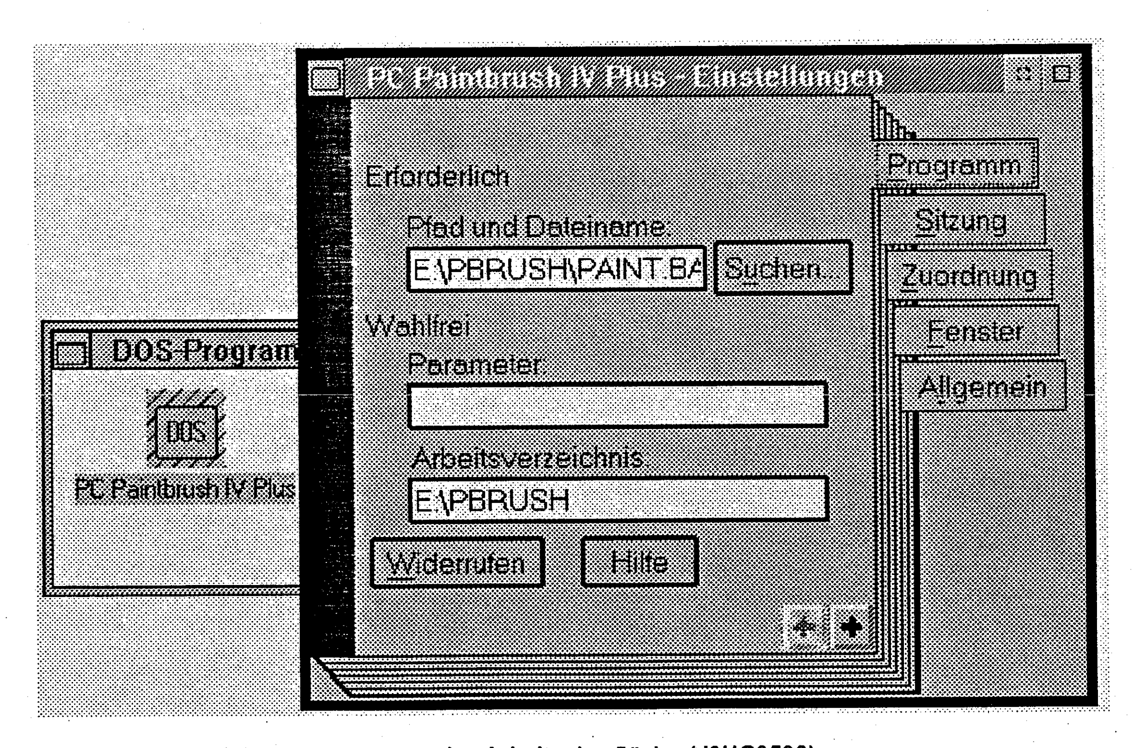


Abbildung 6-3. DOS-Programm an der Arbeitsoberfläche (J8HG0506)

6.4 Speichermanagement unter DOS und WINDOWS

Bedingt durch die schnell zu gering gewordenen Speicherkapazitäten unter DOS hat es eine umfangreiche Entwicklung gegeben mit dem Ziel, die größeren Möglichkeiten der Hardware gegenüber dem Betriebssystem DOS nutzen zu können.
Eine OS/2-Anwendung bedarf natürlich nicht dieser Kniffe, die ja letztlich nichts anderes als gewachsene Notbehelfe sind.
Dieses sind:

- **UMB**, die *Upper Memory Blocks*
- **HMA**, die *High Memory Area*
- **EMS**, die *Expanded Memory Storage*, auch als **LIM 4.0** bezeichnet.
- **XMS**, die *Extended Memory Specification*
- **DPMI**, das *DOS Protected Mode Interface*

Diese „Tricks" werden hier einmal zusammengefaßt behandelt. Denken Sie bitte daran, daß Einstellungen hierfür im Kapitel &doseins. behandelt werden.

6.4.1 Upper Memory Blocks

Die erste Überlegung war die, den Raum des max. konventionellen Speicherbereiches von 640 kB dadurch nutzen zu können, daß andere Teile in den Raum zwischen 640 kB und 1MB verlagert werden, also in die *upper memory blocks*.
Was wird denn nun dort hinein gelegt?

- **TSR's**, also *terminate and stay resident*- Routinen
  - Dies geschieht mit einem AUTOEXEC.BAT-Eintrag:

    ```
    loadhigh c:\Pfad\Programmdatei
    ```

    Statt *loadhigh* ist auch die Angabe „lh" zulässig.
- Treiber
  - Dies geschieht mit dem CONFIG.SYS-Eintrag:

    ```
    devicehigh=c:\Pfad\Treiberdatei
    ```

Diese machen durch den Transport in die **UMB's** einen gleichgroßen Raum im unteren Speicherbereich frei. Charakteristisch hierbei ist, daß es sich um eine statische Lösung handelt, also nicht während einer DOS-Sitzung verändert werden kann.

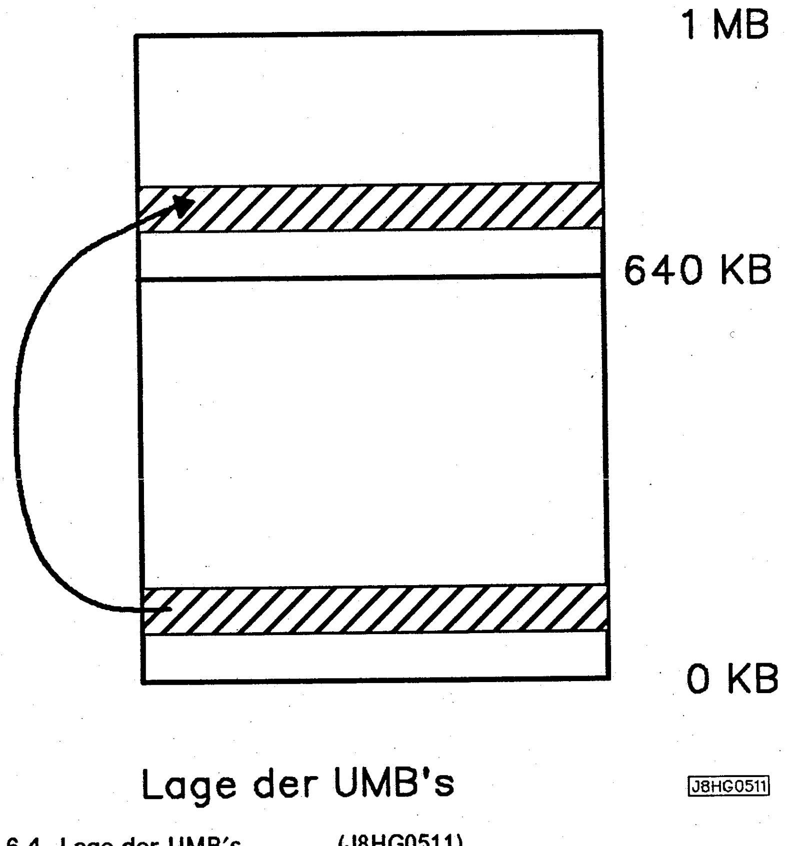


Abbildung 6-4. Lage der UMB's (J8HG0511)

6.4.2 High Memory Area

Der nächste Trick ist ein ganz ähnlicher. Er ist insofern hardwareabhängig, als der anzusprechende Bereich erst ab dem Prozessor 80286 zur Verfügung steht. Bei dieser Methode wird der nächste 64 kB - Bereich oberhalb der 1MB-Grenze angesprochen.
Genutzt wird dieser Bereich mithilfe des CONFIG.SYS-Eintrags

```
dos=high
```

Dann wird der größte Teil des Betriebssystems DOS in die HMA geladen und macht entsprechend Arbeitsspeicher im konventionellen Bereich frei.

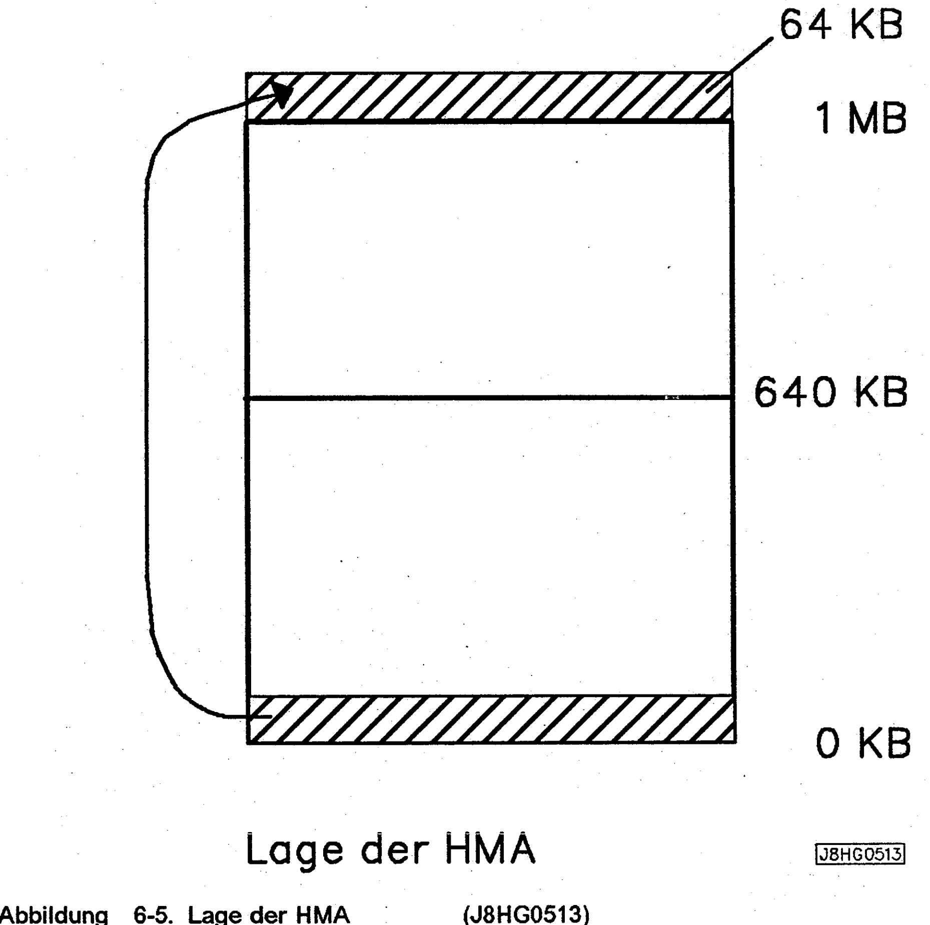


Abbildung 6-5. Lage der HMA (J8HG0513)

6.4.3 Expanded Memory Storage

Bei dieser Methode handelt es sich um eine, die nicht von DOS selbst genutzt wird, aber von LOTUS 1-2-3. Im Gegensatz zum **UMB** ist diese Methode dynamisch. Die Arbeitsweise ist völlig anders, als die oben dargestellten Methoden. Es werden Bereiche aus dem Bereich oberhalb 1 MB in den Bereich zwischen 640 kB und 1 MB gespiegelt. Dies funktioniert allerdings nur, wenn dort zusammenhängende *frames* von 64 kB zur Verfügung stehen. Daher kann es erforderlich sein, daß Sie von Hand die freien Bereiche so zusammenstellen, daß Sie jeweils 4 Fenster mit je 16 kB zu einem Block von 64 kB zusammenstellen. Beim IBM PS/2 kann dies ganz einfach mit der Referenzdiskette erreicht werden, während es bei anderen Maschinen mit Schaltern gemacht werden muß.
Wichtig:
Bei Tokenring- und SCSI-Adaptern ist **EMS** nicht möglich, da der Platz bereits durch diese belegt ist.

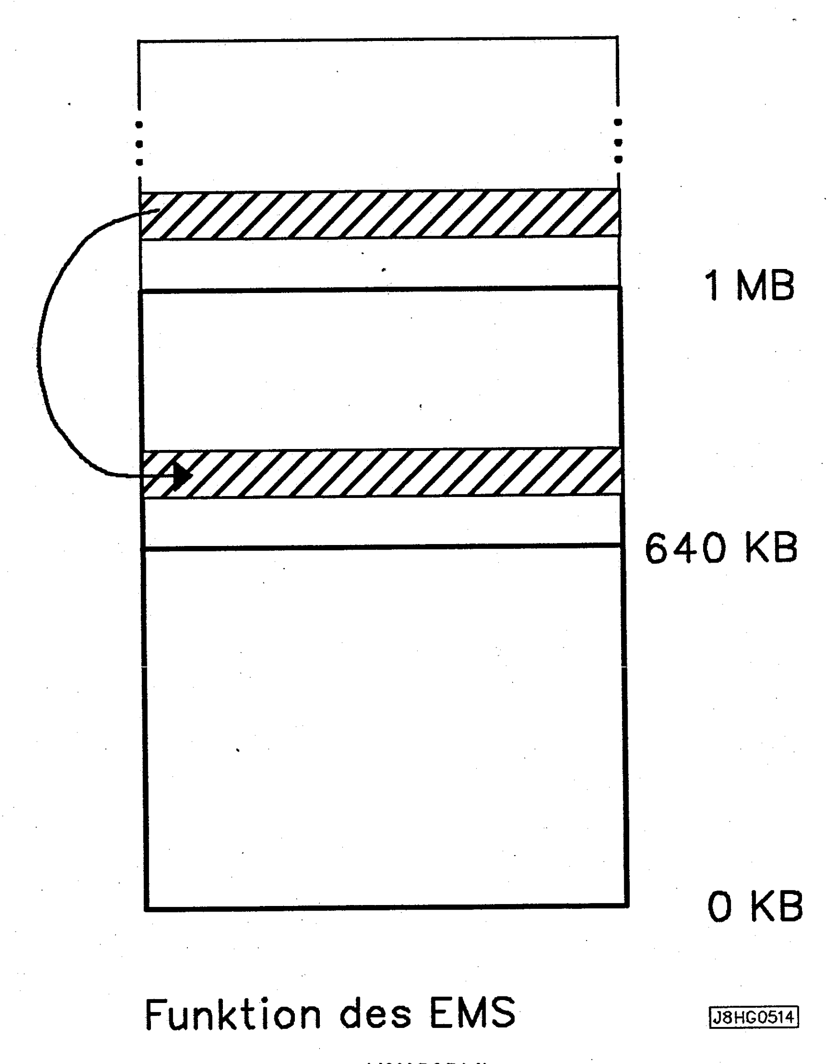


Abbildung 6-6. Funktion des EMS (J8HG0514)

6.4.4 Extended Memory Specification

Dies ist nun die Methode, die den ganzen Bereich oberhalb 1 MB zugänglich macht. WINDOWS nutzt diese Methode. Der CONFIG.SYS-Eintrag:

```
device=himem.sys
```

ermöglicht den Zugriff.

6.4.5 DOS Protected Mode Interface

Diese Methode wurde ebenfalls für WINDOWS 3.0 entwickelt. Es handelt sich um den Versuch, *Preemptive Multitasking* unter WINDOWS durchzuführen und hat in WINDOWS den **Enhanced Mode** zur Folge. Nur sehr wenige Anwendungen existieren für diesen Mode.

6.4.6 DOS-Einstellungen

In der CONFIG.SYS können Device-Treiber und andere Programme für alle VDM-Sitzungen geladen werden, gleiches gilt auch für andere DOS-Einstellungen (Umgebungsvariablen u.ä.). Bestimmte virtuelle Gerätetreiber müssen in der CONFIG.SYS geladen werden, damit die DOS-Sitzungen in OS/2 Version 2 gestartet werden können.

Mit den DOS-Einstellungen können für jedes DOS-Objekt auf der Arbeitsoberfläche spezifische Veränderungen und Anpassungen vorgenommen werden. Dazu sind über das Kontextmenü des Objektes die Einstellungen..., Sitzungen, DOS-Einstellungen... zu öffnen.

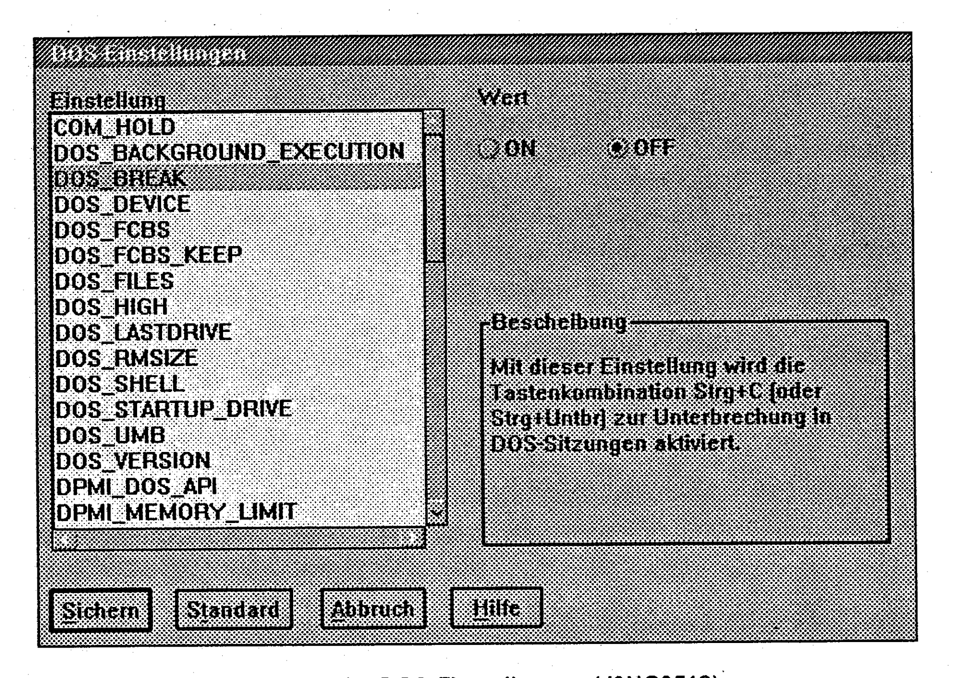


Abbildung 6-7. Dialogfenster für die DOS-Einstellungen (J8HG0512)

Behandlung der DOS-Einstellungen

1. Allgemeine DOS-Einstellungen:

- **DOS_BACKGROUND_EXECUTION**: Mit „OFF“ können Sie steuern, daß eine Anwendung stehen bleibt, wenn Sie nicht im Vordergrund steht. Sie erzielen damit u.U. einen Performance-Gewinn. Der Standardwert ist „ON“.
- **DOS_BREAK:** „ON“ ermöglicht die Programmunterbrechung mit „STRG + C“ oder „STRG + UNTBR“. Der Standardwert ist „OFF“.
- **DOS-LASTDRIVE:** Definition der *lastdrive*. Jeder zusätzliche Laufwerksbuchstabe belegt 100 bytes. Der Standardwert ist „Z“.
- **DOS-SHELL:** Angabe, wo der DOS-Befehlsprozessor steht. Der Standard ist: C:\OS2\MDOS\COMMAND.COM C:OS2\MDOS /P. Die geltenden Parameter sind DOS-identisch.
- **DOS-VERSION:** Simulation einer DOS (OS/2) Version für einzelne Programme (Versionsnr., Unterversnr., Anzahl der Simulationen),
- **MOUSE_EXCLUSIVE_ACCESS:** Ein evtl. zweiter Mauszeiger in einer DOS/WINDOWS-Sitzung kann mit „ON“ abgeschaltet werden.
- **DOS_AUTOEXEC:** Hier können Pfad und die DOS-Startdatei definiert werden.

2. Speicher-Definitionen

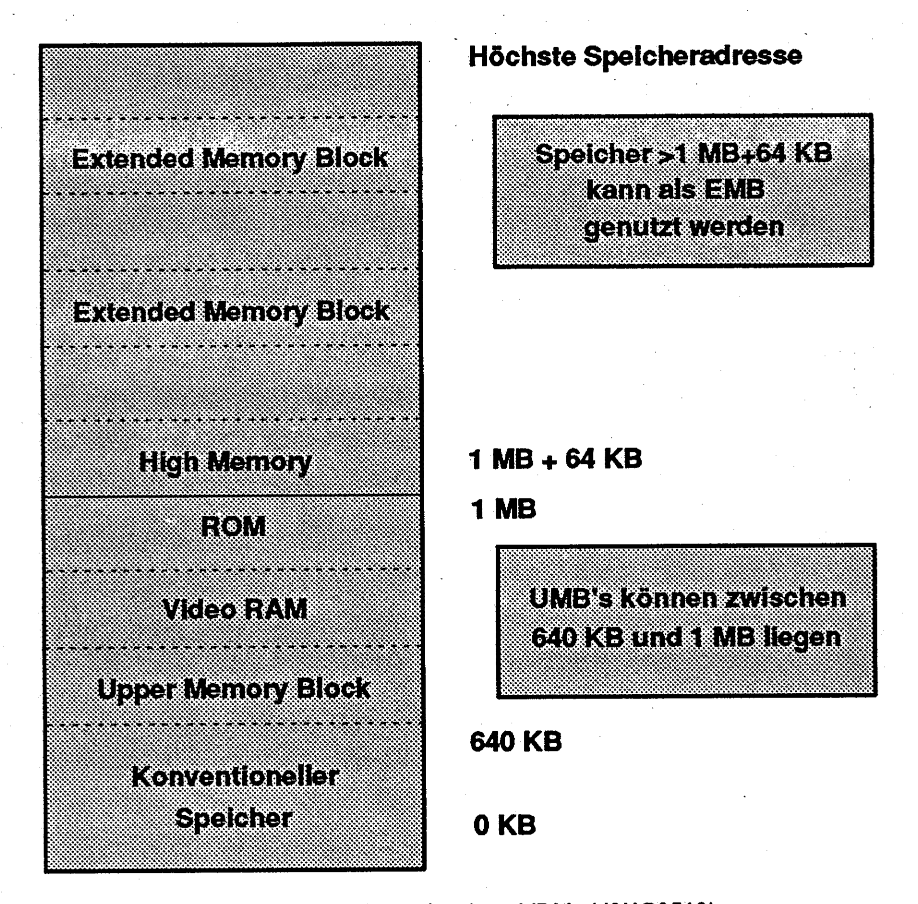


Abbildung 6-8. Speicherverwaltung in einer VDM (J8HG0510)

- **DOS_FCBS:**Angabe der max. *File Control Blocks*. Der Standardwert ist 16. Möglich sind Werte zwischen 0 und 255.
- **DOS_FCBS_KEEP:** Angabe der min. *File Control Blocks*. Diese können vom Betriebssystem nicht geschlossen werden. Der Standardwert ist 8. Möglich sind Werte zwischen 0 und 255.
- **DOS_FILES:** Angabe der max. Anzahl der Dateikennungen in einer Sitzung. Der Standardwert ist 20. Möglich sind Werte zwischen 20 und 255.
- **DOS_HIGH:** Mit „ON“ wird der DOS-Kernel in den Bereich oberhalb der 640 KB geladen und stellt somit mehr konventionellen Speicher zur Verfügung. Der Standardwert ist „OFF“.
- **DOS_RMSIZE:** Definition der Größe des konventionellen DOS-Speichers. Der Standardwert ist 640 kB. Möglich sind Werte zwischen 128 und 640 kB.

- **DOS_UMB:** Ermöglicht die Benutzung der *upper memory blocks* für Treiber und TSR's.
- **MEM_EXCLUDE_REGIONS:** Angabe von Bereichen (mit HEX-Adresse), die nicht vom EMS und XMS angesprochen werden sollen. Das ist erforderlich, wenn bestimmte Bereiche für Einheitentreiber benötigt werden.
- **MEM_INCLUDE_REGIONS:** Angabe von Bereichen, die (innerhalb des Bereiches zwischen 640 kB und 1 MB) von EMS und XMS verwendet werden.
- **XMS_HANDLES:** Angabe der Anzahl interner Dateikennungen für XMS. Standard ist 32, möglich zwischen 0 und 128. Da jede definierte Menge von Dateikennungen Speicherplatz belegt, sollte nur soviel gesetzt werden, wie der Dokumentation der Anwendung zu entnehmen ist.
- **XMS_MEMORY_LIMIT:** Definition des Extended Memory für die einzelne DOS-Sitzung. Der Standardwert ist 2048 kB, möglich ist zwischen 0 und 16384 kB.
- **XMS_MINIMUM_HMA:** Feineinstellung des Extended Memory. Der Nutzen liegt darin, daß Sie das Extended Memory auf das erforderliche Maß reduzieren können.
- **DPMI_DOS_API:** Speicheradressen, die oberhalb 1 MB, also im Protected Mode Speicher stehen, können in Adressen unterhalb 1 MB geändert werden.
- **DPMI_MEMORY_LIMIT:** Definition des *Protected Mode Memory*. Der Standardwert ist 2, der maximale Wert ist 512 MB.
- **DPMI_NETWORK_BUFF_SIZE:** Definition des Übersetzungspuffers für Netzwerk für DPMI- Anwendungen. Der Standardwert ist 8 kB, möglich sind Werte zwischen 1 und 64 kB.
- **EMS_FRAME_LOCATION:** Die Lage des EMS-Frames kann definiert werden. Der Standardwert ist „AUTO".
- **EMS_HIGH_OS_MAP_REGION:** Anpassung des EMS-Bereiches. Der Standardwert ist 32, das Maximum ist 96 kB.
- **EMS_LOW_OS_MAP_REGION:** EMS - Einstellungen für Anwendungen, die den konventionellen Speicher neu adressieren können. Der Standard-Wert ist 384, möglich sind Werte zwischen 0 und 576 kB.

- **EMS_MEMORY_LIMIT:** Definition der für die Anwendung zur Verfügung stehenden EMS-Größe. Der Standard ist 4, möglich sind Werte zwischen 0 und 32 MB.

3. Bildschirm/-Adapter-Einstellungen

- **VIDEO_8514A_XGA_IOTRAP:** Mit „OFF" wird schneller Zugriff auf einen vorhandenen 8514A-Adapter ermöglicht. Wichtig, die Zwischenablage kann dann nicht verwendet werden. Evtl. treten beim Umschalten zwischen Anwendungen Probleme mit dem Bildaufbau auf. Dem kann mit der „VIDEO_SWITCH_NOTIFICATION" begegnet werden.

- **VIDEO_FASTPASTE:** „ON" erhöht die Übermittlungsgeschwindigkeit aus der Zwischenablage. Wichtig, dies funktioniert nicht bei Anwendungen, die Tastatur-Interrupts direkt überwachen.

- **VIDEO_MODE_RESTRICTION:** Durch Einschränkung auf die Video-Modi CGA oder Mono wird der Speicherbereich des höerwertigen Video-Adapters freigesetzt. Dadurch steht der Anwendung mehr Speicher zur Verfügung. Dies ist nicht bei Anwendungen sinnvoll, die mit hohen Auflösungen arbeiten.

- **VIDEO_ONDEMAND_MEMORY:** Benötigt, wenn eine Gesamtbildschirm-Anwendung in der höchsten Auflösung laufen soll. Dies hat zur Folge, daß der Gesamtbildschirm in einem reservierten Speicherbereich steht, somit ist dieser sehr schnell wieder aufgebaut. Bei „OFF" wird der Speicherplatz erst beim Schalten in die Sitzung zur Verfügung gestellt. Der Bildaufbau ist dann langsamer.

- **VIDEO_RETRACE_EMULATION:** Definition der Bildschirmrückverfolgung. Bei „ON" ist der Bildaufbau zwar u.U. langsamer, aber zuverlässiger. Bei „OFF" wird das Bild schneller wieder aufgebaut, kann aber u.U. zu einem leeren Bildschirm führen, da die Rückverfolgung nur in dem Intervall stattfindet, das durch die DOS-Anwendung vorgegeben wird.

- **VIDEO_ROM_EMULATION:** Funktionen wie

  ```
  WRITECHAR
  WRITETTY
  ```

  und Rollen eines Bildschirmes werden nicht von ROM der Video-Karte, sondern vom Betriebssystem ausgeführt und sind somit schneller.

- **VIDEO_SWITCH_NOTIFICATION:** Nur sinnvoll im Zusammenhang mit den Einstellungen „VIDEO-8514A_XGA_IOTRAP" und „VIDEO_ONDEMAND_MEMORY". „ON" wird benötigt, wenn der OS/2-Bildschirmtreiber nicht alle vom Bildschirmadapter zur Verfügung gestellten Funktionen unterstützt. Der Standard ist „OFF".

- **VIDEO_WINDOW_REFRESH:** Definition der Zeit, nach der ein Fenster neu angezeigt bzw. in den Bildschirmpuffer gestellt wird. Standardwert ist 1 entsprechend 0,1 Sekunden,

4. Hardware- und besondere Einstellungen

- **AUDIO_ADAPTER_SHARING** Zugriffssteuerung auf den Audio- Adapter. Diese Funktion steht nur nach Installation der Multimedia-Unterstützung zur Verfügung.

- **COM_HOLD**: reserviert den Zugriff auf serielle Schnittstellen für die jeweilige VDM. Der Standardwert ist „OFF".

- **DOS-DEVICE:** Ein DOS-Einheitentreiber kann geladen werden. Wichtig ist die Pfadangabe, z.B.:

  C:\OS2\MDOS\ANSI.SYS

- **PRINT_TIMEOUT:** Definition der Zeit zwischen dem Anstoß des Druckjobs und Weiterleitung an den Drucker. Diese Einstellung hat große Bedeutung bei Erzeugung von sehr großen Druckjobs, die u.U. sonst in mehrer kleine Druckjobs aufgeteilt würden. Der Standardwert ist 15 sek., möglich zwischen 0 und 3600 sek..

- **HW_NOSOUND:** Mit „ON" kann der Lautsprecher abgeschaltet werden. Standard ist „OFF".

- **HW_ROM_TO_RAM:** Das Betriebssystem kopiert das *read only memory* in das *random access memory*. Dies kann zum *debuggen* des Kernel benutzt werden.

- **HW_TIMER:** Bei „ON" haben Anwendungen Zugriff auf die 8253 timer ports.

- **IDLE_SECONDS:** Einer Anwendung, die auf Eingabe wartet, kann nach hier definierter Zeit der Prozessorzugriff entzogen werden. Der Standard ist 0, der Maximalwert 60 Sekunden.

- **IDLE_SENSITIVITY:** Nur im Zusammenhang mit IDLE_SECONDS sinnvoll. Dieser Wert wird in % ausgedrückt und stellt den Schwellen-Wert dar, ab dem eine

Anwendung als „IDLE“ angesehen wird. Der Standardwert ist 75%, möglich ist zwischen 1 und 100%.

- **KBD_ALTHOME_BYPASS:** Mit der Einstellung „OFF“ wird ermöglicht, daß eine DOS-Sitzung vom Fenster auf Gesamtbildschirm und umgekehrt mit den Tasten ALT + POS1 umgeschaltet wird. Der Standard ist „OFF“.
- **KBD_BUFFER_EXTEND:** Einstellung des DOS-Befehlspuffers. Der Standardwert ist „ON“ und entspricht einem Puffer von 128 byte. „OFF“ entspricht einem Puffer von 16 byte.
- **KBD_CTRL_BYPASS:** Die Tastenkombinationen STRG + ESC und ALT + ESC können als OS/2-Befehle aufgehoben und der DOS-Sitzung zur Verfügung gestellt werden.
- **KBD_RATE_LOCK:** Bei „ON“ wird verhindert, daß eine Anwendung die Tastaturwiederholrate verändert.
- **COM_SELECT:** Sie können hiermit den Zugriff auf COM ports einschränken.
- **INT_DURING_IO:** Sie können *interrupts* während Schreib- und Leseoperationen zulassen.

5. Pfadangabe für *Virtual Machine Boot*:

- **DOS-STARTUP-DRIVE:** Gilt nur beim Laden einer bestimmten DOS-Version beim VMB und gibt die Lage des aufzurufenden DOS-Kernels an.

6. Spezielle WINDOWS - Einstellung:

- **WIN_RUNMODE:** Einstellung des WINDOWS-Modus. Der Standard ist „AUTO“matisch, bei evtl. Erkennungsschwierigkeiten kann die Einstellung manuell auf den Standard-Mode für Version 3.0 oder REAL-Mode eingestellt werden.
- **WIN_DDE:** Ermöglicht DDE zwischen „öffentlichen“ WIN-OS2- und OS/2-Sitzungen durchzuführen.
- **WIN_CLIPBOARD:** Ermöglicht den Datenaustausch zwischen WIN-OS2- und OS/2-Sitzungen mithilfe des Zwischenspeichers.

6.5 Booten von virtuellen DOS-Maschinen

Das Betriebssystem OS/2 Version 2 stellt die Möglichkeit zur Verfügung im Rahmen der virtuellen DOS-Maschinen ein 8086-Betriebssystem zu booten. Das heißt OS/2 ist in der Lage, mehrere DOS-Versionen gleichzeitig virtuell zu verwalten, egal ob es sich z.B. um IBM DOS 4.0 oder DR-DOS 5.0 usw. handelt. Damit kann der Anwender neben den (emulierten) DOS 5-Sitzungen auch eine originale DOS-Version für bestimmte Anwendungen nutzen.
Unter OS/2 wird dieses Verfahren VMB - *Virtual Machine Boot* genannt.

Es existieren 3 Möglichkeiten, das DOS-Betriebssystem über VMB in eine virtuelle DOS-Maschine zu laden:

1. mittels einer bootfähigen DOS-Startdiskette,
2. über eine Abbilddatei (*Image*) von einer DOS-Diskette, die auf die Festplatte abgespeichert wurde (Befehl: VMDISK),
3. über eine primäre DOS-Partition der Festplatte (nicht mit Dual Boot oder Boot Manager zu verwechseln, letzteres ist aber Voraussetzung).

Hinweise zur Arbeit mit dem VMB

- Mit **FSFILTER.SYS** besteht Zugriff auf HPFS-Platten.
- Überprüfung von CONFIG.SYS und AUTOEXEC.BAT notwendig (z.B. HIMEM.SYS).
- In den DOS-Einstellungen ist in der Regel nur das DOS-STARTUP-DRIVE zu ändern (trifft auch für Abbilddatei-Verfahren zu).
- Analog der Beschränkungen in virtuellen DOS-Maschinen werden auch durch VMB keine VCPI und andere nicht-DPMI-Schnittstellen unterstützt.
- Zu beachten ist, daß bestimmte Sonderfunktionen einer DOS-Version im <u>Virtuellen Maschinen Boot</u> nicht funktionieren, wie z.B. bei DR DOS 6.0 das Superstore.
- *Ill behaved* DOS-Anwendungen werden auch in einem <u>Virtuellen Maschinen Boot</u> Probleme bereiten.

6.5.1 ÜBUNG: Virtuelles Booten eines DOS von Diskette

Ziel der Übung

Sie lernen die Möglichkeit des Virtual Machine Boot von Diskette kennen und anwenden.

Durchführung

1. Es handelt sich um die einfachste Lösung. Öffnen Sie den Ordner Befehlszeilen.
2. Legen Sie die Ihnen vom Dozenten überreichte DOS 3.3 Diskette
   in das Laufwerk A:
3. Aktivieren Sie das Objekt DOS von Laufwerk A: mit Doppelklick.
4. Nachdem die virtuelle DOS-Maschine gestartet wurde, geben Sie im DOS-Prompt bitte

   ver

   ein.

Was können Sie feststellen?

- ______________________________

Denken Sie daran, daß Sie den Mauseintrag in der AUTOEXEC.BAT und den DEVICE-Eintrag für den FSFILTER - Treiber in der CONFIG.SYS vornehmen sollten, wenn Sie auf HPFS zugreifen wollen.

6.5.2 ÜBUNG: Virtuelles Booten eines DOS vom Image

Ziel der Übung

Sie lernen die Möglichkeit kennen, ein DOS von einem Disketten-Image auf einem logischen Laufwerk virtuell zu booten.

Durchführung

Beachten Sie bitte, daß Sie die ersten Schritte hier im Kurs nicht duchführen müssen, da ich Ihnen die Diskette bereits vorbereitet habe. Setzen Sie bitte erst bei Punkt 7 auf.

1. Erstellen Sie eine DOS-Boot-Diskette mit einer DOS-Version Ihrer Erfordernisse.

2. Editieren Sie die AUTOEXEC.BAT entsprechend. In meinem Beispiel habe ich eine DOS 4.0 Version ausgewählt. mit:

```
@ECHO OFF
SET COMSPEC=COMMAND.COM
VERIFY OFF
KEYB GR
ECHO OFF
DATE
TIME
c:\os2\mdos\mouse
VER
```

3. Nun müssen Sie voraussichtlich Ihre DOS-Config.sys editieren. Meine sieht so aus:

```
SHELL=COMMAND.COM /P /E:256
device=fsfilter.sys
COUNTRY=49
FILES=8
INSTALL=NLSFUNC.EXE
LASTDRIVE=e
```

4. Denken Sie bitte daran, den device-Eintrag für den FSFILTER-Treiber als ersten device-Eintrag in die Config.sys zu setzen, sofern Sie beabsichtigen, auf HPFS - formatierte Laufwerke zu zugreifen.

5. Evtl. müssen Sie die lastdrive korrigieren.

6. Damit fsfilter.sys auch aktiviert werden kann, müssen Sie diesen Treiber allerdings auch noch auf die Diskette kopieren. Sie finden ihn im Verzeichnis C:\OS2\MDOS.
7. Nun legen Sie sich auf einem Ihrer Laufwerke einen Ordner z.B. mit dem Namen

   `DOS40`

   an.
8. Nun ein ganz entscheidender Schritt, öffnen Sie eine OS/2-Befehlszeile.
9. Geben Sie ein

   `vmdisk a: d:\dos40\dos40.img`

   Wobei „d" Ihr Ziellaufwerk ist und dos40.img der Name der zu erstellenden Image-Datei.
10. Einen ganz wichtigen Schritt müssen wir noch durchführen. Kopieren Sie das Objekt DOS vom Laufwerk A: mit Kopieren
11. Ändern Sie den Namen dieses neuen Objektes auf z.B.:

    `DOS 4.0 vom Image`
12. Öffnen Sie nun das Kontextmenü des Objektes.
13. Gehen Sie in die Einstellungen.
14. Dann wählen Sie die Seite Sitzung aus.
15. Und nunmehr DOS Einstellungen
16. Jetzt kommen wir der Sache schon wesentlich näher, klicken Sie DOS_STARTUP_DRIVE an.
17. Geben Sie jetzt den kompletten Pfad Ihrer DOS-Image-Datei an.

    `d:\dos40\dos40.img`
18. Jetzt sichern Sie und schließen das Notizbuch.
19. Aktivieren Sie nun Ihr Objekt DOS 4.0 vom Image
20. Geben Sie wieder

    ver

    ein
21. Was stellen Sie fest?

    - ____________________

Dies war ja nun wirklich eine lange Übung. Aber ich hoffe, Sie hat Ihnen den richtigen „VIRTUAL_MACHINE_BOOT_DURCHBLICK“ gebracht.

Für die dritte Möglichkeit ist es erforderlich, daß Sie ein DOS auf einer primären, nicht „hidden" gesetzten Partition haben. Da dies hier auf Ihren Schulungmaschinen nicht zutrifft, zeige ich Ihnen den Weg auf, damit Sie es später am eigenen Arbeitsplatz leichter haben.

- Um diese Möglichkeit zu installieren, Erstellen Sie wieder ein weiteres DOS-Objekt.
- Sie geben hier als STARTUP_DRIVE allerdings hier Ihr Laufwerk

  C:\

  an.
- Denken Sie daran, daß Sie den Mauseintrag in der AUTOEXEC.BAT und den DEVICE-Eintrag für den FSFILTER - Treiber in der DOS-CONFIG.SYS vornehmen sollten, wenn Sie auf HPFS zugreifen wollen.

6.6 Windows-Programme unter OS/2 Version 2

Programme, welche für Windows 3.X geschrieben wurden, lassen sich ohne besondere Vorkehrungen unter OS/2 Version 2 betreiben. Auch dafür ist es erforderlich, bei der Installation die entsprechenden Angaben zu machen. Die Installations-Prozedur erlaubt auch ein automatisches Einbinden vorhandener Windows-Programme in die Oberfläche.

Ansonsten gilt das unter 6.3.2, „Einbinden von DOS-Programmen in die Arbeitsoberfläche" auf Seite 6-8 gesagte für das Anlegen von Objekten in der Arbeitsoberfläche.

6.6.1 Einrichten der Windows-Umgebung

Wenn vor der Installation von OS/2 schon Windows-Programme vorhanden waren, so kann das Installationsprogramm diese finden und in einen Ordner auf der Arbeitsoberfläche einfügen. Die installierende Person kann auswählen, welche Programme auf diese Art an der Arbeitsoberfläche erscheinen sollen.

Später installierte Programme können auf verschiedene Art in die gewünschte Umgebung eingebunden werden:

1. Automatisches Einbinden in den Windows-Programm-Manager durch das Installationsprogramm,
2. Erstellen einer Referenz-Kopie wie bei OS/2-Programmen,
3. Anlegen eines Programm-Objektes mit anschließender Anpassung.

6.6.2 Datenaustausch zwischen OS/2-Programmen und Windows-Programmen

Für den Datenaustausch zwischen Programmen gibt es auf der Benutzerebene:

1. Zwischenablage,
2. Dynamischen Datenaustausch (*Dynamic Data Exchange* = DDE)

Der Datenaustausch zwischen Programmen, welche für verschiedene Plattformen mit verschiedenen Voraussetzungen und Architekturen geschrieben wurden, ist unter OS/2 Version 2 möglich. Durch die Differenzen jedoch gibt es immer wieder „Verständigungsprobleme", welche nicht am OS/2 Version 2 liegen, sondern in den Anwendungsprogrammen. So kann es durchaus sein, daß ein Windows-Programm einen Text in die (Windows-) Zwischenablage stellt, ein OS/2-Programm das gleiche tut, aber kein Programm kann die Daten des anderen bekommen, weil die verwendeten Formate zu unterschiedlich sind.

Die Zwischenablage ist ein Speicherbereich, welcher den Anwendungen als gemeinsamer Bereich zur Verfügung steht. Dieser Bereich kann zu jeder Zeit von jeder Anwendung beschrieben werden. Daher ist die Zwischenablage für nur **ein** Objekt zu einer Zeit vorgesehen. Es gibt drei Standard-Formate:

1. Text
2. Bitmaps
3. OS/2 Meta-Dateien

Anwendungen können jedes beliebige eigene Format eintragen. Das steht dann aber nur den „verständigen" zur Verfügung.

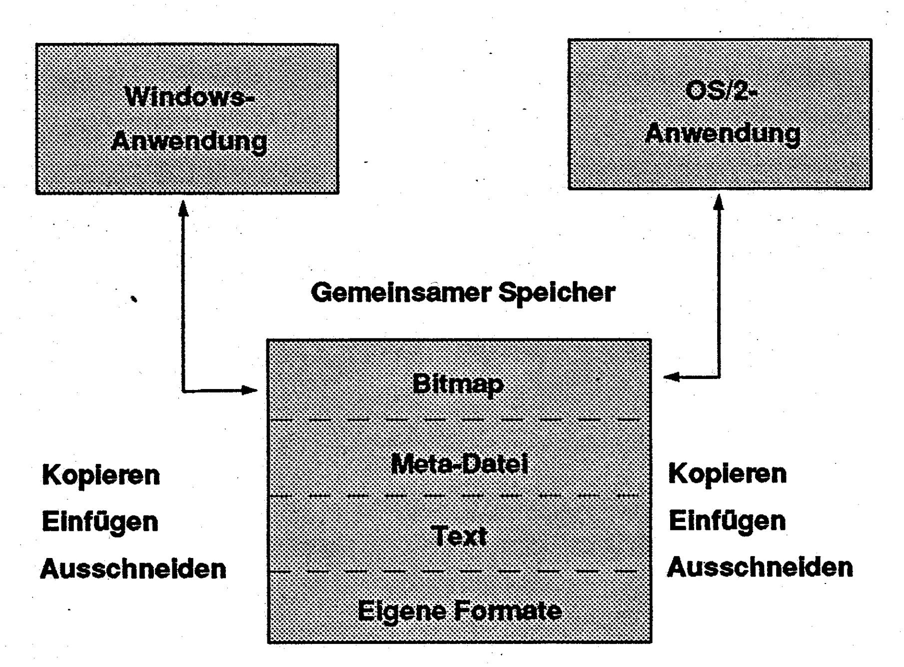


Abbildung 6-9. Konzept der Zwischenablage (J8HG0508)

Das Konzept des Datenaustausches mit DDE ist in Abb. 6-10 auf Seite 6-30 dargestellt. Aus Anwendungssicht ist es fast völlig gleich, welche Art von Programm am anderen Ende sitzt, solange dieses Programm die DDE-Konversation (DDE-Protokoll) beherrscht.

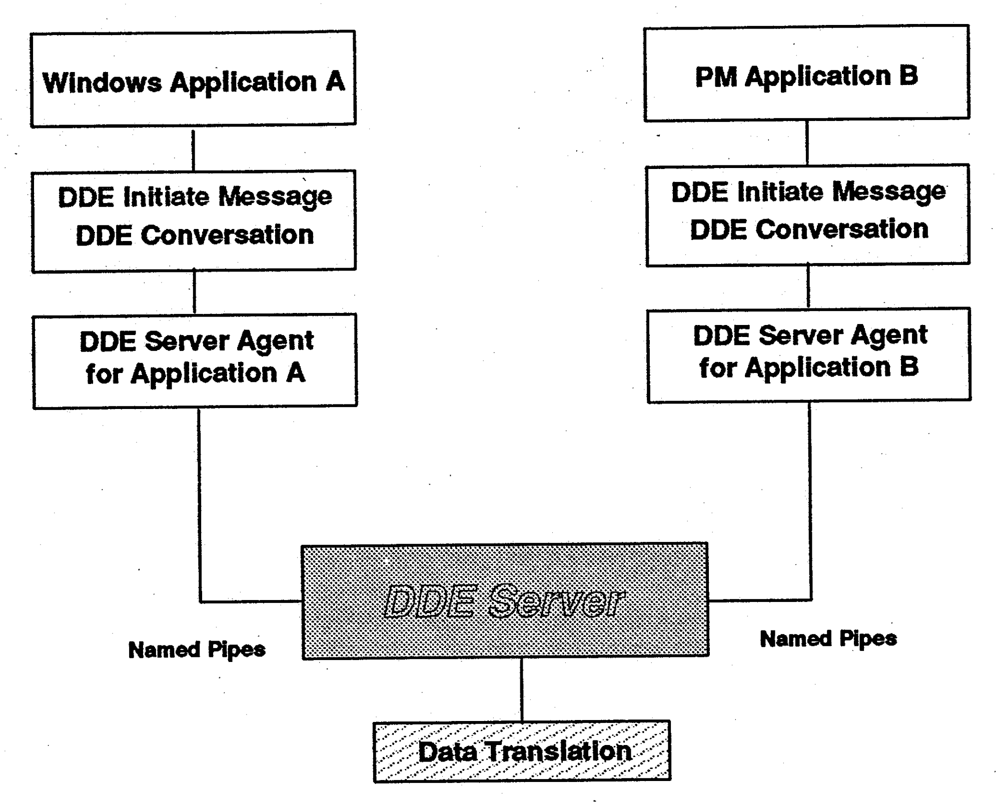


Abbildung 6-10. Konzept des DDE (J8HG0540)

6.7 Installation von Windows-Programmen

Unter DOS-Windows werden Programme oft über den Datei-Manager installiert. Dazu wird der Inhalt der Installationsdiskette angezeigt und dann mit einem Doppelklick das Installationsprogramm aktiviert.

Das ist unter OS/2 Version 2 nicht viel anders:

- In der Arbeitsoberfläche den Inhalt der Installationsdiskette anzeigen,
- Doppelklick auf dem Installationsprogramm,
- OS/2 Version 2 stellt selbst fest, daß es sich um ein Windows-Programm handelt und öffnet eine Windows-Sitzung,
- Die Installation verläuft im übrigen genau so, wie es unter DOS-Windows der Fall wäre.

Kapitel 7. Systemanpassung

Ziele dieses Kapitels

In diesem Kapitel lernen Sie:

- Wie Sie Optimierungen vornehmen können
- Wie mit Hilfe der Datei CONFIG.SYS Einfluß genommen werden kann
- Wie Sie verschiedene Diagnosen durchführen können.

Inhalt dieses Kapitels

7.1 Allgemeines . . . 7-3
7.1.1 Übung: Wiederherstellen der Basisoberfläche . . . 7-5
7.2 Systemoptimierung . . . 7-7
7.3 Die Datei CONFIG.SYS . . . 7-10
7.4 Diagnose . . . 7-16
7.4.1 Allgemeines . . . 7-16
7.4.2 Dienstprogramm zum Prüfen der Integrität von Disketten und Festplatten . . . 7-17
7.4.3 Übung: Das Dienstprogramm CHKDSK . . . 7-17
7.4.4 Systemfehlerprotokoll . . . 7-18
7.4.5 Übung: Das Dienstprogramm SYSLOG . . . 7-19
7.5 Dienstprogramme zum Erstellen von Speicherauszügen . . . 7-20
7.6 Die Dienstprogramme TRACE und TRACEFMT . . . 7-23
7.6.1 Übung: Das Dienstprogramm TRACE . . . 7-26
7.7 Das Dienstprogramm PSTAT . . . 7-27
7.8 Das Dienstprogramm PATCH und die Korrekturdienste . . . 7-28
7.9 Anzeige des Änderungsstandes des OS/2 . . . 7-29
7.10 Tips und Hinweise zu OS/2 Version 2 . . . 7-30
7.10.1 Übung: README-Überblick . . . 7-31
7.10.2 Übung: Betriebssystemstart bei Schwierigkeiten 7-33

7.1 Allgemeines

Bei der Installation des Betriebssystemes wurden Voraussetzungen für einen guten Gebrauch des Systemes geschaffen. Die bei der Installation festgelegten Parameter mögen sich jedoch später als nicht optimal herausstellen oder es werden tiefgreifende Änderungen vorgenommen. Diese gilt es dann mit der Systemanpassung zu berücksichtigen.

Ein Teil der Systemanpassung wird mit einem (oder mehreren) Dienstprogramm(en) durchgeführt, je nachdem, welche Änderungen erforderlich sind. Einige allgemeine Einstellungen werden mit der Systemkonfiguration, *(System Setup)* gemacht.

Die Benutzung der Systemanpassung führt zu einer Anzeige oder Änderung der Dateien OS2.INI und OS2SYS.INI, welche sich im Verzeichnis OS2 befinden. Diese Dateien haben ein gepacktes Datenformat und sind daher nur von Programmen aus oder durch Benutzung der Systemanpassung veränderbar.

Die hier abgespeicherten Werte werden als Standardwerte beim Laden von OS/2 benutzt. Sie sind jedoch jederzeit änderbar, mit der Systemanpassung, von einem Programm aus oder auch mit OS/2-Befehlen. Weiterhin können alle Programme auf die *.INI-Dateien zugreifen, um die Voreinstellungen abzufragen und zu benutzen. Beispiel: Die Information über das gewünschte Datumsformat kann abgefragt und ein Datum entsprechend ausgegeben werden.

Systemeinstellungen, die sich optisch auf Anwendungen und Darstellungen auswirken, also z.B. Fenstergrößen, Farbwerte, Lage der Objekte, Rahmenbreiten werden in der Datei OS2.INI festgehalten.

Die Auswahl Logo-Anzeige, System, Logo, sowie installierte Schriftarten und Druckertreiber schlagen sich dagegen in der Datei OS2SYS.INI nieder.

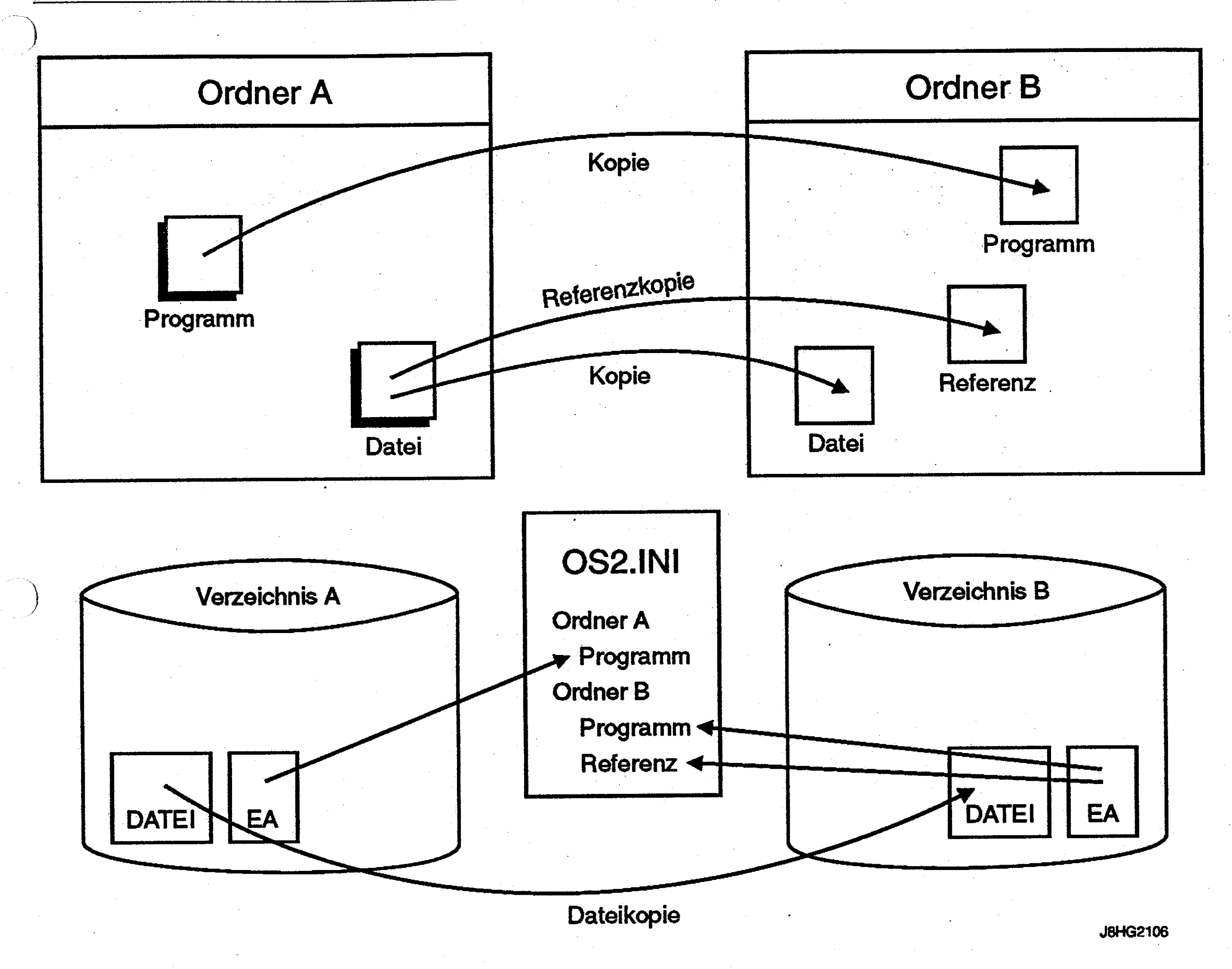


Abbildung 7-1. OS2.INI und Referenz (J8HG2106)

7.1.1 Übung: Wiederherstellen der Basisoberfläche

Ziel der Übung

Sie erfahren, wie Sie auf einfache Weise sich eine bereits angepaßte OS/2-Oberfläche wiederherstellen können

Durchführung

1. Führen Sie einen ordnungsgemäßen Systemabschluß durch
2. Booten Sie das System neu
3. Sofort nach dem Bootmanager oder beim Kaltstart nachdem der Speichertest durchgelaufen ist, drücken Sie die Tastenkombination ALT + F1
4. Halten Sie die Tasten solange bis Ihr PS/2 „melodische" Töne von sich gibt, dann lassen Sie die Tasten los.

Als Ergebnis dieser Aktion sollten Sie nun eine neue Arbeitsoberfläche vor sich haben, die bereits die Ordner der vorher integrierten beinhaltet.

Was passiert bei diesem Prozess?

- OS/2 erstellt Ihnen eine neue Arbeitsoberfläche:
  - Die OS2.INI aus dem Verzeichnis OS2 wird mit dem Namen OS2.001 umkopiert.
  - Die OS2SYS.INI aus dem Verzeichnis OS2 wird mit dem Namen OS2SYS.001 umkopiert.
  - Die CONFIG.SYS wird mit dem Namen CONFIG.001 umkopiert.
  - Die entsprechenden Dateien aus dem Verzeichnis OS2\INSTALL werden Ihrer Hardware entsprechend und Ihrer vorigen Arbeitsoberfläche annähernd angepaßt und in das Verzeichnis OS2 kopiert.
- Nun wird genau wie bei der Grundinstallation, die Arbeitsober fläche ganz neu aufgebaut. Sie werden feststellen, daß der Prozeß länger dauert, als ein normaler Systemstart.
- Nachdem der Aufbau abgeschlossen ist und das System startet, können Sie feststellen, daß die Struktur der Arbeitsoberfläche auf Ihrer Festplatte zweimal existiert.

- Wenn Sie die Maschine mit dem Mini-OS/2 booten, können Sie die ältere löschen.
- Wichtig ist in dem Zusammenhang, daß mit diesem Prozeß Ihre alte selbst beeinflußte Oberfläche nicht mehr vorhanden ist.

7.2 Systemoptimierung

Durch Veränderung von Standard-Einträgen in der CONFIG.SYS können Sie das Verhalten und die Leistung des OS/2 verändern und ggf. verbessern. Beachten Sie bitte, daß es hier zwischen den verschiedenen Eingriffsmöglichkeiten Abhängigkeiten gibt. Wenn Sie im Zweifel über die Bedeutung und Auswirkung sind, ist es immer empfehlenswert, das Online-Referenzhandbuch zu Rate zu ziehen. Sollten Sie den gewünschten Begriff nicht gleich finden, steigen Sie mit dem Suchbegriff CONFIG.SYS in die Hilfe ein.

MAXWAIT

Dieser Wert sagt aus, nach welcher Zeit OS/2 spätestens die Priorität von threads erhöht, wenn dem Thread keine Zeitscheibeneinheit zugeteilt wurde. Dies stellt sicher, daß jede der gestarteten Anwendungen auch bedient wird. Der bei der Basisinstallation gesetzte Standardwert beträgt 3 sek. Sie können den Wert zwischen 1 und 255 Sekunden variieren. WICHTIG, dies ist abhängig von der Einstellung PRIORITY.

TIMESLICE

Sie können hiermit die Zeitscheiben (Minimalwert/Maximalwert) definieren, die den threads zur Verfügung gestellt werden. Der Systemstandard ist dynamisch, was in der Regel die beste Performance ergibt. Die Werte können manuell zwischen 32 und 65536 Millisekunden umdefiniert werden.

PRIORITY

OS/2 setzt standardmäßig den Parameter dynamic, was zur Folge hat, daß Prioritäten dynamisch verändert werden. Somit wird sichergestellt, daß JEDE Anwendung Prozessorzeit zugeordnet bekommt.

THREADS

Diese Angabe sagt aus, wieviel THREADS gleichzeitig aktiv sein können. Der vom System gesetzte Minimalwert ist mit 64 definiert. Der Maximalwert kann 4095 betragen.

SWAPPATH

OS/2 definiert bei der Installation den Pfad OS2\SYSTEM, zu empfehlen ist, einen eigenen **Swappath** zu definieren, den Sie sogar auf ein dafür reserviertes logisches Laufwerk legen. Als minfree können Sie zwischen 512 und 32767kB definieren. Bei Unterschreitung des definierten Wertes erfolgt eine Warnung. Abhängigkeit besteht zum Eintrag **MEMMAN**.

MEMMAN

In diesem Statement wird das Memory management mittels **Swap** ermöglicht oder auch abgeschaltet. In der Regel wird die bessere Performance mit **MEMMAN=SWAP** erreicht. Ausnahme, Einsatz einer zeitabhängigen Anwendung.

CACHE

Die Nutzung des **Cache** bewirkt schnelleren Datenzugriff, da ganze Blocks in den Arbeitsspeicher gestellt werden und somit weniger zeitaufwendige Plattenzugriffe erfolgen müssen. Der **Cache** arbeitet nach dem Prinzip der **least recent used** Blocks. Es handelt sich hierbei um den HPFS-Cache.

DISKCACHE

Entsprechend dem **Cache** für HPFS - Platten, wird der **Diskcache** bei FAT - Platten wirksam.

BUFFERS

Dieser Eintrag drückt die Zahl der anzulegenden **Buffers** aus. Ein Buffer belegt 512 bytes. Je mehr **Buffers** Sie anlegen, desto schneller wird Ihr System durch weniger Plattenzugriffe. ABER: Ihr verfügbarer Arbeitsspeicher wird dadurch kleiner, was Ihre Performance wieder hinunterdrücken kann.

7.3 Die Datei CONFIG.SYS

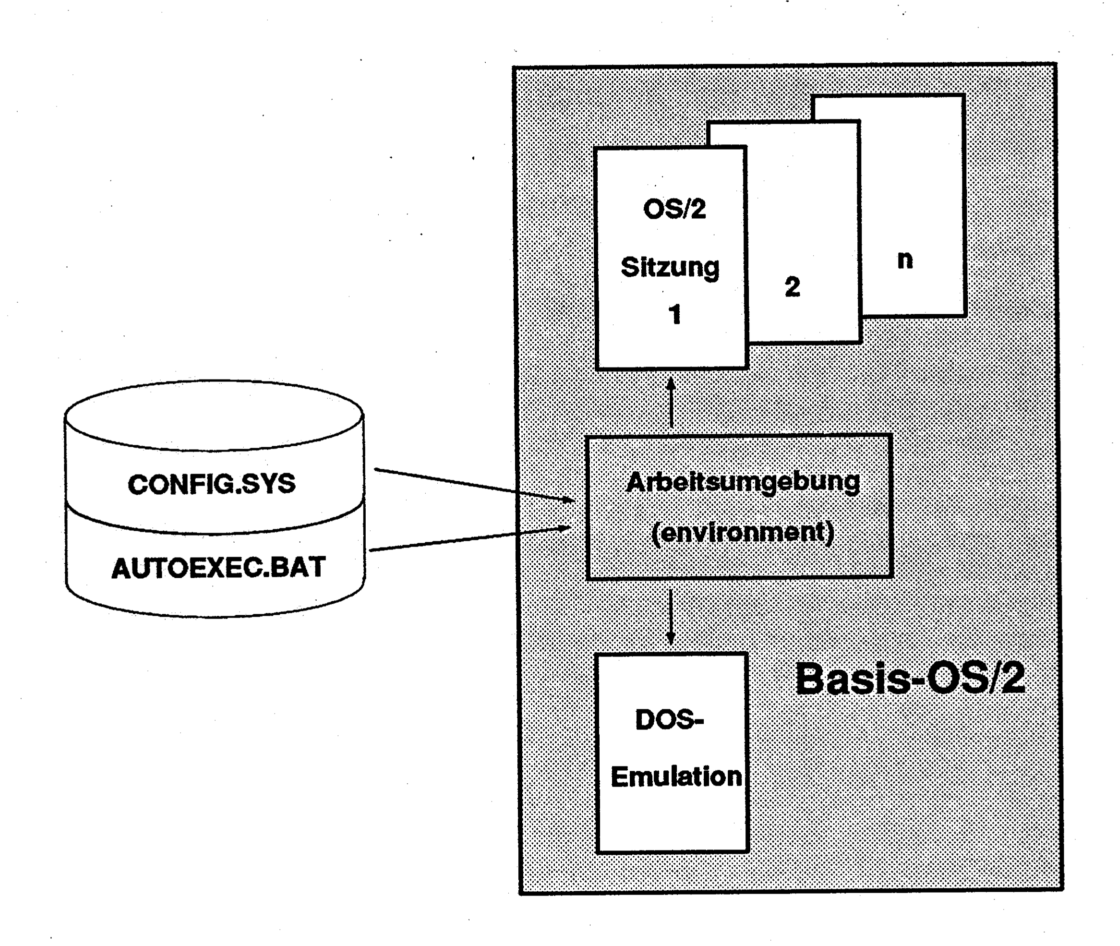


Abbildung 7-2. Geltungsbereich der CONFIG.SYS (J8HG0602)

```
REM HINWEIS: BEI BESONDEREM INTERESSE FINDEN SIE DETAILS IM
REM REFERENZHANDBUCH UND IM DOS HANDBUCH

REM FEHLERHINWEIS, BEIM DURCHLAUFEN DER CONFIG.SYS
REM ER SOLLTE AM ANFANG STEHEN
PAUSEONERROR=YES

REM HIER WIRD DER TREIBER FÜR DAS HPFS GELADEN MIT 512 KB CACHE UND
REM CACHE-RECORDSATZ VON 4 KB, MIT AUTOMATISCHEM PRüFEN DER HPFS-PLATTEN
IFS=C:\OS2\HPFS.IFS /CACHE:512 LAZY:ON /CRECL:4 /AUTOCHECK:FGJK

REM START EINER PROGRAMMROUTINE WÄHREND DER INITIALISIERUNG,
REM DIE OS2.INI WIRD GESICHERT
REM Sicherung der Arbeitsoberfläche (deutsch und HPFS)
REM dazu müssen folgende Unterverzeichnisse angelegt werden:
REM \SICHARBO + \SICHARBO\S1 + \SICHARBO\S2
REM in SICHARBO befindet sich die Datei "RETTARBO.CMD"
REM Hinweis: bei FAT muß "Arbeitsoberfläche" durch
REM                      "Arbeitso"   ersetzt werden.
REM Hier beginnt die Sicherung
call=\os2\xcopy.exe \sicharbo\*.in1 \sicharbo\*.in2
call=\os2\xcopy.exe \os2\*.ini \sicharbo\*.in1
call=\os2\xcopy.exe \sicharbo\s1\*.* \sicharbo\s2\*.* /s /e
call=\os2\xcopy.exe "\Arbeitsoberfläche\*.*" \sicharbo\s1\*.* /s /e
call=\os2\eautil.exe "\Arbeitsoberfläche" \sicharbo\s1\desktop.ea /s /r /p
REM Ende der Sicherung der Arbeitsoberfläche

REM ANGABE DER BENUTZERSCHNITTSTELLEN
PROTSHELL=C:\OS2\PMSHELL.EXE
SET USER_INI=C:\OS2\OS2.INI
SET SYSTEM_INI=C:\OS2\OS2SYS.INI
SET OS2_SHELL=C:\OS2\CMD.EXE

REM WICHTIG, HIERMIT WIRD DIE OBERFLÄCHE DEFINIERT
SET AUTOSTART=PROGRAMS,TASKLIST,FOLDERS
SET RUNWORKPLACE=C:\OS2\PMSHELL.EXE

REM ANGABE DES BEFEHLSPROZESSORS
SET COMSPEC=C:\OS2\CMD.EXE

REM SUCHPFAD FüR AUSFüHRBARE PROGRAMME
REM .EXE, .COM,  .CMD (.BAT, für DOS)
REM SET SETZT EINE ZEICHENKETTE FÜR SPÄTERE VERWENDUNG
REM HIER WERDEN DIE PFADE FÜR GENANNTE PROGRAMME GESETZT
SET PATH=C:\BIN;C:\OS2;C:\OS2\SYSTEM;C:\MUGLIB;F:\SQLLIB;F:\CMLIB;
F:\IBMLAN\NETPROG;G:\BIN;G:\DEVEL\EASEL-EE;C:\OS2\INSTALL;C:\;
G:\TOOLKT13\BIN;G:\OS2UTIL;F:\APPL\OIS;C:\OS2TAPE;F:\SIDEKICK;F:\PMWORD;

REM DATENPFAD
REM SET SETZT EINE ZEICHENKETTE FÜR SPÄTERE VERWENDUNG
REM HIER WERDEN DIE PFADE FÜR DATENPFADE GESETZT
SET DPATH=C:\OS2;C:\OS2\SYSTEM;C:\MUGLIB\DLL;F:\CMLIB;F:\IBMLAN\NETPROG;
G:\LPEX;G:\DEVEL\EASEL-EE;G:\EASEL-EE\PICTURES;C:\OS2\INSTALL;C:\;
G:\TOOLKT13\IPFC;G:\OS2UTIL;F:\APPL\OIS;
```

```
REM SUCHPFAD FüR DYNAMIC LINK LIBRARIES;
REM DER PUNKT AM PFADBEGINN ERLAUBT SUCHEN NACH DLL IN EIGENEM
REM VERZEICHNIS
LIBPATH=.;C:\OS2\DLL;C:\DLL;G:\DLL;G:\DEVEL\EASEL-EE\DLL;F:\APPL\123G;
G:\DEVEL\ENFIN;C:\MUGLIB\DLL;F:\SQLLIB\DLL;F:\CMLIB\DLL;F:\IBMLAN\NETLIB;
C:\;G:\OS2UTIL;F:\APPL\OIS\DLL;C:\OS2TAPE;G:\TOOLKT13\DLL;F:\TX4-2;F:\PMWORD;
REM PFAD FüR HILFSTEXTE
SET HELP=G:\LPEX;C:\OS2\HELP;G:\TOOLKT13\BIN;F:\APPL\OIS\HELP;G:\TOOLS\CUADRAW;
REM AUFRUF DES GLOSSARY
SET GLOSSARY=C:\OS2\HELP\GLOSS;

REM GIBT DER VORDERGRUNDSESSION PRIORITÄT BEI PLATTENZUGRIFFEN
PRIORITY_DISK_IO=YES

REM ANGABE DES OS2 PROMPTS
SET PROMPT=$I[$P]GUTEN MORGEN

REM SUCHPFADE FüR INFORMATIONSDATEIEN U. DEFINITION PROGREF-KOMPONENTEN
SET BOOKSHELF=C:\OS2\BOOK;G:\TOOLKT20\PROGREF;

REM DIE REFERENZHANDBÜCHER KÖNNEN MITEINANDER KOMBINIERT WERDEN
SET PROGREF=PRINTRO.INF+PRCP.INF+PRGPI.INF+PRWIN.INF+PRDATA.INF+EX1.INF

REM **** ANGABEN FüR IBMC2 UND EASEL
REM SET INCLUDE=F:\IBMC2\INCLUDE;F:\DEVEL\EASEL-EE
REM SET LIB=C:\LIB;F:\IBMC2\LIB;C:\OS2

REM  >>>>>  ANGABEN, DIE SICH AUF DIE PERFORMANCE AUSWIRKEN  <<<<<
REM ANGABE MAX WARTEZEIT FÜR PROGRAMME MIT STANDARDPRIORITÄT
REM IN SEKUNDEN
MAXWAIT=3
REM ANGABE MIN UND MAX WERTE FÜR ZEITSCHEIBEN IN MILLISEKUNDEN
TIMESLICE=45,125
REM ABLAUFSTEUERUNG FÜR THREADS, DYNAMIC ODER ABSOLUTE
PRIORITY=DYNAMIC
REM ANGABE DER MAX. AKTIVEN VON EINANDER UNABHÄNGIGEN FUNKTIONEN
THREADS=255
REM BESTIMMUNG DES PFADES FÜR SWAP, MINFREE-ANGABE AUF DER
REM PLATTE IN KB, MAX IST 32767
SWAPPATH=J:\ 512
REM DER BENUTZTE SPEICHERBEREICH KANN AUF EINE PLATTE
REM AUSGELAGERT WERDEN, APIs KÖNNEN AUF GESCHÜTZTE SPEICHERBEREICHE
REM ZUGREIFEN
MEMMAN=SWAP,PROTECT
REM DEFINITION DES FAT-PLATTENCACHE MIT LazyWrite
DISKCACHE=1024,LW
REM DIE ANGABE DER BUFFERMENGE
BUFFERS=60

REM ALLGEMEINE ANGABEN FüR OS/2
IOPL=YES
SET KEYS=ON
```

```
REM HIER WERDEN DIE VERZEICHNISSE FÜR ZU LÖSCHENDE DATEIEN GESETZT
REM DIE DAFÜR MAX. Z. VERFÜGUNG STEHENDE GRÖSSE
SET DELDIR=C:\DELETE,512;D:\DELETE,512;E:\DELETE,512;F:\DELETE,512;G:\DELETE,512;

REM LADEN VON BASIS-EINHEITENTREIBERN
BASEDEV=PRINT02.SYS
BASEDEV=IBM2FLPY.ADD
BASEDEV=IBM2SCSI.ADD /LED
BASEDEV=OS2SCSI.DMD
BASEDEV=OS2DASD.DMD

REM >>>>>>            UND HIER DIE DOS - UMGEBUNG          <<<<<<
REM ERWEITERTE TASTATURFUNKTIONEN (DOS)
REM DEVICE=C:\OS2\ANSI.SYS
REM AUSWAHL DES BETRIEBSMODUS, NUR PROTECTED ODER AUCH REAL
REM FÜR DOS
PROTECTONLY=NO

REM LADEN VON DOS EINHEITENTREIBERN INS UPPER MEMORY
DEVICEHIGH=C:\OS2\MDOS\ANSI.SYS
REM ALLGEMEINE ANGABEN FüR DOS; COMMAND.COM IST
REM DOS-EMULATION

REM INSTALLATION DES BEFEHLSPROZESSORS, HIER SPEICHERRESIDENT
SHELL=C:\OS2\MDOS\COMMAND.COM C:\OS2\MDOS /P
REM MÖGLICHKEIT DER PROGRAMMUNTERBRECHUNG MIT STRG UND UNTBR
BREAK=ON
REM VERWENDUNG VON DATEISTEUERBLÖCKEN,File Control Blocks
FCBS=16,8
REM GRÖßE DES FÜR DOS ZUR VERFÜGUNG STEHENDEN SPEICHERPLATZES
RMSIZE=640
REM DEFINITION DER LAGE DES DOS-KERNEL, HIER INNERHALB DER 640 KB
REM UND DOS  BENUTZT NICHT UPPER MEMORY BLOCKS
DOS=LOW,NOUMB

REM FÜR VDM WERDEN VIRTUELLE EINHEITENTREIBER BENÖTIGT:
REM HIER DER VIRTUAL EXPANDED MEMORY MANAGER
DEVICE=C:\OS2\MDOS\VEMM.SYS
REM DANN DER VIRTUAL MOUSE DRIVER
DEVICE=C:\OS2\MDOS\VMOUSE.SYS
REM NUN DER VIRTUAL DOS PROTECT MODE EXTENDER
DEVICE=C:\OS2\MDOS\VDPX.SYS
REM DANACH DER VIRTUAL EXTENDED MEMORY SPECIFICATION TREIBER
DEVICE=C:\OS2\MDOS\VXMS.SYS /UMB
REM VIRTUAL DOS PROTECT MODE INTERFACE
DEVICE=C:\OS2\MDOS\VDPMI.SYS
REM VIRTUAL WINDOWS TREIBER
DEVICE=C:\OS2\MDOS\VWIN.SYS
REM HIER DER DDE-SERVER FÜR WINDOWS
RUN=C:\OS2\MDOS\WINOS2\VDMSRVR.EXE
DEVICE=C:\OS2\MDOS\VVGA.SYS
DEVICE=C:\OS2\MDOS\VCOM.SYS
```

```
REM VORBEREITUNG DES VGA-BILDSCHIRMS FÜR ZEICHENUMSETZTABELLEN
DEVINFO=SCR,VGA,C:\OS2\VIOTBL.DCP
REM AUSSERDEM MÖGLICH FÜR TASTATUR UND DRUCKER
REM WEITERE ANGABEN ZUR BENUTZUNG DES VGA
SET VIDEO_DEVICES=VIO_IBMVGA
SET VIO_IBMVGA=DEVICE(BVHVGA)
REM HIER DIE ENTSPRECHENDEN ANGABEN ZUR BENUTZUNG DES 8514/A
REM DEVINFO=SCR,BGA,C:\OS2\VIOTBL.DCP
REM SET VIDEO_DEVICES=VIO_IBM8514A
REM SET VIO_IBM8514A=DEVICE(BVHVGA,BVH8514A)

REM LADEN VON ERFORDERLICHEN EINHEITENTREIBERN
DEVICE=C:\OS2\DOS.SYS
DEVICE=C:\OS2\IBMMOU02.SYS
DEVICE=C:\OS2\MOUSE.SYS TYPE=IBMMOU$
DEVICE=C:\OS2\PMDD.SYS
DEVICE=C:\OS2\POINTDD.SYS
DEVICE=C:\OS2\PCLOGIC.SYS SERIAL=COM2
DEVICE=C:\OS2\COM.SYS

REM NLS-ANGABEN
REM SPEZIFIZIERT DEN LÄNDERBEZOGENEN SPRACHENCODE
COUNTRY=049,C:\OS2\SYSTEM\COUNTRY.SYS
REM MÖGLICHKEIT DER DATUMSBESTÄTIGUNG UND ÄNDERUNG
REM DATE TT.MM.JJ
REM MÖGLICHKEIT DER ZEITBESTÄTIGUNG UND ÄNDERUNG
REM TIME HH.MM
REM VORBEREITUNG DER TASTATUR FÜR ZEICHENUMSETZTABELLEN
DEVINFO=KBD,GR,C:\OS2\KEYBOARD.DCP

REM DEFINIERUNG DER GÜLTIGEN CODEPAGES
CODEPAGE=850,437

REM ANGABEN ZUR BENUTZUNG DES QUERY MANAGER
SET QRWDR=F:
SET QRWINST=F:\SQLLIB

REM ANGABEN ZUR BENUTZUNG HELP MANAGER
SET IPFC=G:\TOOLKT20\IPFC;

REM TREIBER UND VARIABLE FüR ANWENDUNGEN
REM **** ANGABEN FüR LPEX UND C/2
SET LPATH=G:\LPEX;G:\BIN;G:\LPEX\EXTRAS
REM SET TMP=F:\IBMC2\TMP

REM TREIBER FüR BANDGERÄT UND START VON AUTOTAPE
DEVICE=C:\PMTAPE\IP2IDRV.SYS IROPT=I

REM START EINER PROGRAMMROUTINE WÄREND DER INITIALISIERUNG
REM EMPFEHLUNG: DEN DEVICE EINTRAG DAVOR SETZEN
RUN=C:\PMTAPE\ITDMN.EXE
```

```
REM PROTOKOLLIEREN VON SYSTEMEREIGNISSEN, PARAMETER SIEHE
REM BEFEHLSREFERENZ
REM DAZU WIRD DIE LOG-FILE AUFGERUFEN, IN DIESEM BEISPIEL EIN
REM BUFFER VON 32kB ANGELEGT, MIN IST 4, MAX IST 64kB
DEVICE=C:\OS2\LOG.SYS /E:32
RUN=C:\OS2\SYSTEM\LOGDAEM.EXE /E:C:\OS2\SYSTEM\LOG0001.DAT

REM ANLEGEN VON DRUCKERBUFFERN AB VERSION 2.x
REM HIER: HIER FÜR LPT1 BIS LPT3
PRINTMONBUFSIZE=2048,134,134
```

7.4 Diagnose

7.4.1 Allgemeines

Das Betriebssystem OS/2 bietet verschiedene Möglichkeiten der Ablaufverfolgung und der Fehlerbestimmung. Hierzu gehören folgende Dienstprogramme:

- CHKDSK - Prüfen der Integrität von Disketten und Laufwerken
- PSTAT - Information über laufendene Prozesse und deren Resourcen
- SYSLEVEL - Anzeige des CSD - Levels (Servicestufe) von OS/2
- TRACE - Systemablaufverfolgung OS/2 (nach Systemereignissen)
- SYSLOG - Fehleraufzeichnung im Systemprotokoll (error log)
- CREATEDD - Diskettenformatierung für Speicherauszug (Dump)
- PATCH - Dienstprogramm zur Korrektur von Softwarefehlern

Mit diesen Dienstprogrammen können die unter dem Begriff **RAS** zusammengefaßten Systemmerkmale:

- Reliability = Zuverlässigkeit
- Availability = Verfügbarkeit
- Serviceability = Servicefreundlichkeit

gewährleistet werden. Diese Werkzeuge sollten nur mit Hilfe des technischen Koordinators bzw. OS/2-Administrators ausgeführt werden.

7.4.2 Dienstprogramm zum Prüfen der Integrität von Disketten und Festplatten

Bei einem Systemabbruch kann es passieren, daß auf Ihren Platten nicht komplett zurückgeschriebene, also fragmentierte Dateien stehen. Um dieses erkennen und ggbf. reparieren zu können, können Sie das Dienstprogramm *CHKDSK* benutzen. In der objektorientierten Umgebung gibt es neben den bekannten Möglichkeiten der Befehlseingabe in einer Befehlszeile auch eine objektbezogene im Objekt Laufwerke *(Drives)*.

7.4.3 Übung: Das Dienstprogramm CHKDSK

Ziel der Übung:

In diesem Abschnitt üben Sie den Umgang mit CHKDSK resp. PMCHKDSK

Die Übungsschritte:

1. Öffnen Sie das Objekt Laufwerke
2. Wählen Sie eines der dort gezeigten logischen Laufwerke aus
3. Drücken Sie die Manipulationstaste der Maus
4. Wählen Sie Datenträger prüfen... mit Doppelklick der Selektionstaste

7.4.4 Systemfehlerprotokoll

Um das Systemprotokoll nutzen zu können, muß der zugehörige Einheitentreiber für die Systemprotokollierung in der Datei CONFIG.SYS gestartet werden (*OS/2 Error Log Facility*). Z.B. mit folgender Anweisung:

```
DEVICE=C:\OS2\LOG.SYS /E:32
```

Zusätzlich zu dieser Anweisung muß der Dämon-Prozeß für die Protokollierung von Systemfehlern aktiviert werden. Hiermit wird ein Plattenbereich angelegt, in welchem Systemfehler des OS/2 aufgezeichnet werden. Dies ist mit dem Befehl RUN in der CONFIG.SYS möglich:

```
RUN=C:\OS2\SYSTEM\LOGDAEM.EXE /E:C:\OS2\SYSTEM\LOG0001.DAT
```

Standard für die Systemprotokolldatei ist die Bezeichnung LOG0001.DAT und eine Größe von 64KB, diese Werte sind änderbar.

Um den Inhalt der Protokolldatei in formatierter Form zur Anzeige zu bringen muß das Kommando:
SYSLOG
aufgerufen werden. Ohne Parameter wird ein PM-Programm mit entsprechendem Standardfenster aufgerufen.
Das Protokollieren von Einträgen in der Systemprotokolldatei kann unterbrochen bzw. wiederaufgenommen werden (Parameter /S /R).

Wenn zur Basis-Version des OS/2 zusätzlich die Erweiterten Service-Funktionen installiert werden, wird die LOG-Fehleraufzeichnung automatisch mitinstalliert, um Fehler des CM und des DBM aufzuzeichnen.

7.4.5 Übung: Das Dienstprogramm SYSLOG

Ziel der Übung:

In diesem Abschnitt üben Sie den Umgang mit dem System-Log.

Die Übungsschritte:

1. Aktivieren Sie eine Befehlszeile.
2. Stellen sie fest, ob die Log-Funktion initialisiert wurde.
3. Lassen sie sich das Systemlog anzeigen, indem Sie in der Befehlszeile
   SYSLOG
   eingeben.

7.5 Dienstprogramme zum Erstellen von Speicherauszügen

Wenn ein Fehler sich nur schwer analysieren läßt (z.B. „Absturz des Systems"), kann es erforderlich werden, den Hauptspeicherinhalt auf Diskette abzuziehen. Eine Analyse kann nur von entsprechenden IBM Stellen vorgenommen werden. Es ist darauf zu achten, daß durch den Speicherauszug keine vertraulichen Daten oder lizensierte Informationen an IBM gegeben werden.

Erforderlich ist dazu mindestens eine **Dump-Diskette**, welche mit dem Dienstprogramm CREATEDD erstellt werden muß. CREATEDD formatiert eine Diskette 720 KB oder 1,4 MB und kopiert das Ladeprogramm zum Starten und ein Programm zum Kopieren des Speicherinhaltes auf Diskette.
Die erforderliche Eingabe ist:

```
CREATEDD A:
```

Mit diesem Befehl kann eine Speicherauszugsdiskette (Dump-Diskette) für das vom Betriebssystem unabhängige Auswertungsprogramm für Speicherauszüge erstellt werden. Ein Service-Koordinator sollte immer eine solche Diskette vorrätig haben. Werden für einen Speicherauszug mehrere Disketten benötigt, muß die erste Diskette mit CREATEDD vorbereitet werden; die restlichen Disketten können beliebige formatierte Disketten sein.

Beim Starten des Auswertungsprogramms werden alle Systemvorgänge gestoppt und nur noch das Auswertungsprogramm wird fortgesetzt.

Das Auswertungsprogramm kann entweder manuell oder über das Dienstprogramm TRAPDUMP gestartet werden.

Manuelles Starten

Der Speicherauszug wird durch die Tasten-Kombination Ctrl-Alt-Num-Num angefordert (die Tasten Strg und Alt gedrückt halten und die Taste Num zweimal drücken). Nach einigen Sekunden wird der Bildschirm gelöscht, und die folgende Nachricht wird angezeigt:

```
Die mit CREATEDD erstellte Diskette für den
Speicherauszug (Dump) in Laufwerk A einlegen
und eine beliebige Taste drücken.
```

Es folgt die Nachricht:

```
Der Speicherauszug (Dump) wird erstellt ...
```

Bei Bedarf werden Sie zum Diskettenwechsel aufgefordert. Bitte Beachten Sie, daß der Speicherabzug nicht einfach abgebrochen werden kann!

Beim Beenden des Speicherauszugs erscheint folgende Nachricht:

```
Diskette Nummer 1 einlegen, um den Speicherauszug (Dump)
zu beenden.
```

Dadurch kann vom Steuerprogramm der Summensatz für den Speicherauszug geschrieben und der Speicherauszug beendet werden.
Nach Beendigung des Vorgangs werden die folgenden Nachrichten angezeigt:

```
Der Speicherauszug (Dump) ist beendet. Die Dump-Diskette aus
dem Laufwerk entnehmen und das System neu starten.
```

Merke:

Das Dump-Programm kann nicht aus einer OS/2-Sitzung heraus gestartet werden, auch nicht von einer virtuellen DOS-Maschine aus. Wird das DUMP-Programm mit einem Warm-Start geladen, so kopiert es automatisch den Speicherinhalt auf die Diskette wie mit Ctrl-Alt-Num-Num. Da jedoch zuvor die Speicherteste abliefen, dürfte ein solcher Speicherauszug fragwürdig sein.

Dienstprogramm TRAPDUMP:

Warnung:

Die Aktivierung von TRAPDUMP ermöglicht dem System jedes Mal automatisch die Initialisierung des unabhängigen Auswertungsprogramms für Speicherauszüge, wenn ein Fehler im Anwendungscode auftritt. Das Programm TRAPDUMP sollte jedoch nur dann aktiviert werden, wenn das Betriebssystem auf Fehler geprüft werden muß und vom technischen Koordinator eine entsprechende Anweisung ergangen ist.

Zum Aktivieren von TRAPDUMP in der Datei CONFIG.SYS folgende Anweisung angeben:

```
TRAPDUMP=ON
```

Durch die Angabe dieser Anweisung wird von TRAPDUMP automatisch das unabhängige Auswertungsprogramm für Speicherauszüge initialisiert, wenn ein Fehler auftritt, durch den die Anwendung normalerweise beendet wird (z. B. Allgemeiner Zugriffsfehler (*general protection fault*) - 000D).

7.6 Die Dienstprogramme TRACE und TRACEFMT

Diese Programme helfen bei der Rekonstruktion eines Ablaufes, welcher zu einem Fehler geführt hat. Mit der Trace-Funktion können Sequenzen von Systemereignissen, Funktionsaufrufen oder Daten protokolliert werden. Das Protokoll wird in der Regel zu Testhilfezwecken erstellt. Sind die gewünschten Daten gesammelt, können sie mit der Formatierfunktion aus dem Trace-Puffer abgerufen und formatiert an den Bildschirm, Drucker oder eine Datei gesendet werden.

Eine solche Ablaufverfolgung sollte nur zu Diagnosezwecken eingesetzt werden. Eine sehr gute Kenntnis des OS/2 ist unabdingbar. Da die Aufzeichnung von Systemereignissen Zeit kostet, können Programme in ihrem Laufzeitverhalten beeinflußt werden.

Ein TRACE wird durch einen Eintrag in CONFIG.SYS ermöglicht:

```
TRACE=ON
```

Der Pufferspeicher für die Systemereignisse ist 4KB groß, kann aber mit

```
TRACEBUF=n
```

auf einen Wert bis zu 63KB festgelegt werden. Es muß einer der beiden Anweisungen in der CONFIG.SYS stehen, um die Ablaufverfolgung aktivieren zu können.

Eine Beschreibung der Systemereignisse befindet sich im Referenzhandbuch der OS/2-Kommandos.

Beispiele für die Aktivierung von TRACE

```
TRACE ON KERNEL
```

Mit dem Schlüsselwort KERNEL werden alle dynamischen Ablaufverfolgungspunkte aktiviert. Die dynamischen Ablaufverfolgungspunkte sind als Einträge in TDF-Dateien (*trace definition file*) implementiert.

```
TRACE ON /P:34
```

Mit der Option P: wird die Ablaufverfolgung für die Prozeß-ID mit dem hexadezimalen Wert 0034 aktiviert. Es wäre auch die Angabe /P:ALL möglich.

```
TRACE /C /R
```

Der Puffer für die Ereignisablaufverfolgung wird gelöscht, und anschließend wird die Ereignisablaufverfolgung wiederaufgenommen.

Da ein **TRACE** mit einer Vielzahl von parametergesteuerten Varianten durchgeführt werden kann, ziehen Sie im Bedarfsfall das Referenzhandbuch zu Rate.

Formatierte Anzeige der Trace-Daten

Mit dem Dienstprogramm TRACEFMT können die formatierten Datensätze der Ablaufverfolgung in der umgekehrten Zeitfolge angezeigt werden, in der sie aufgezeichnet wurden.
Mit Hilfe des Dienstprogramms TRACEFMT werden die während der Systemablaufverfolgung in den Trace-Puffer gestellten Daten formatiert. TRACEFMT ist eine Präsentations-Manager-Anwendung, die in einem Fenster ausgeführt wird. Demzufolge bietet das Programm eine gute Hilfeunterstützung. In der Menüleiste für diese Anwendung stehen mehrere Auswahlmöglichkeiten zur Verfügung.
Mit Hilfe dieser Auswahlmöglichkeiten können folgende Funktionen durchgeführt werden:

- Öffnen und Sichern der formatierten und/oder sortierten Daten in einer Datei
- Abrufen der Daten im Puffer
- Drucken der nach Prozeß-ID oder Ereigniscode sortierten Daten
- Anzeigen formatierter Daten als Übersicht

Das Dienstprogramm TRACEFMT kann so oft verwendet werden, bis der Fehler gefunden wurde; ein Neustart des Systems ist nicht erforderlich.

7.6.1 Übung: Das Dienstprogramm TRACE

Ziel der Übung:

In diesem Abschnitt üben Sie den Umgang mit dem Dienstprogramm TRACE

Die Übungsschritte:

1. Aktivieren Sie den OS/2-Systemeditor
2. Suchen Sie sich die Datei CONFIG.SYS
3. Nehmen Sie folgende Einträge vor:

```
TRACE=ON
TRACEBUF=8
```

4. Sichern Sie die CONFIG.SYS
5. Führen Sie einen Systemabschluß durch und booten Sie erneut.
6. Aktivieren Sie eine OS/2 Befehlszeile
7. Geben Sie ein:
   Tracefmt
8. Aus dem sich öffnenden Fenster heraus können Sie bei Bedarf Ihr Trace-Ergebnis auf den Drucker legen.

7.7 Das Dienstprogramm PSTAT

PSTAT zeigt den Zustand einer Reihe von System-Ressourcen an. Damit lassen sich Informationen einholen, welche Sitzungen, Prozesse, unabhängigen Prozesse *(threads)* usw. zu dem Zeitpunkt des Aufrufes aktiv sind.

PSTAT wird mit einem oder mehreren Parametern aufgerufen:

- /C - Prozesse und Sitzungen, Identifikation und Zusammengehörigkeit
- /S - Semaphore und ihre Benutzer
- /L - Anzeige der Laufzeit-Bibliotheken und den benutzenden Prozessen
- /M - Information über gemeinsame Speicherbenutzung
- /P:PID - Anzeige aller obigen Informationen für einen ausgewählten Prozess (PID = Process-ID in Hex-Darstellung)

Mit dem Befehl PSTAT kann auch festgestellt werden, warum ein bestimmter Thread den Status „*blocked*" hat (auf ein Systemereignis wartet) oder warum die Leistung eines Threads gering ist (niedrige Priorität im Vergleich zu anderen Threads). Zudem wird die Prozeß-ID angezeigt, die von jedem Prozeß zugeordnet wird. Die Prozeß-ID kann dann als Eingabe für das Dienstprogramm TRACE (Ablaufverfolgung auf Prozeßbasis) verwendet werden.

Eine weitere wichtige Information ist die Anzeige der von einem Programm benutzten DLL's.

7.8 Das Dienstprogramm PATCH und die Korrekturdienste

Bei einer komplexen und umfangreichen Software kommt es unvermeidlich zu Fehlern oder Unzulänglichkeiten. Wenn diese Fehler vom Labor behoben wurden, dann werden „Austauschmodule“ oder kleinere Korrekturen verfügbar gemacht. Diese werden über DIAL IBM oder über HONE auf ein PS/2 heruntergeladen und können dann

- Über die vorhandenen Module kopiert werden.
- Mit PATCH zur Korrektur verwendet werden.

Da häufig nicht erkennbar ist, ob überhaupt ein Fehler vorliegt, oder wenn ja, wo der Fehler begründet ist, müssen die Anweisungen, welche in README.TXT mit den Austauschmodulen kommen, beachtet werden.

Das Programm PATCH vergleicht vor jeder Änderung den vorhandenen Stand des OS/2 bzw. der betroffenen Komponente mit dem Stand dessen, was auf einer Patch-Diskette vorhanden ist.

Der Benutzer kann selbst zu jeder Zeit mit dem Befehl SYSLEVEL den vorhandenen Stand abfragen. Es werden alle Komponenten mit ihren individuellen Daten ausgewiesen. Die Disketten tragen die Bezeichnung **CSD = Corrective Service Diskettes**, ihr Inhalt ist kumulativ aufgebaut, was bedeutet, daß u.U. mehrere Disketten benötigt werden.

7.9 Anzeige des Änderungsstandes des OS/2

Oftmals ist es erforderlich, sich einen Überblick über das Betriebssystem und seine Komponenten zu verschaffen.
Dazu dient der Befehl: **SYSLEVEL**

Mit diesem Befehl werden alle Dateien des OS/2 durchsucht und das Ergebnis wird angezeigt. Es kann einige Minuten dauern, bis das Ergebnis angezeigt wird.

7.10 Tips und Hinweise zu OS/2 Version 2

Merke:

Grundsätzlich sollte die auf den Installationsdisketten vorhandene Datei „README“ (Diskette 10) studiert werden. Hier liegt diese Datei im gepackten Format vor und wird beim Installationsprozeß entpackt. Sie steht dann als ASCII-Datei zur Verfügung.

- Mauszeiger ist auf der Arbeitsoberfläche nicht verfügbar, und Sie möchten trotzdem einen ordnungsgemäßen Systemabschluß durchführen.
  Reaktion: Grundsätzlich sollte immer zunächst ein Warmstart versucht werden, bevor der Stromschalter betätigt wird, weil beim Warmstart der OS/2-Kernel vor dem IPL zuerst alle Systempuffer zurückschreibt.
  Bessere Reaktion: Mit Strg + Esc zur Fensterliste umschalten, die OS/2 - Arbeitsoberfläche mit Cursor-Tasten und ENTER auswählen. Falls auf dem Desktop Objekte ausgewählt sind, diese Auswahl mit Leertaste aufheben. Nun kann mit SHIFT + F10 das Kontextmenü aufgerufen werden, und der Systemabschluß angewählt werden.

7.10.1 Übung: README-Überblick

Ziel der Übung

Sie machen sich mit der README vertraut und erfahren welche Themenbereiche dort behandelt werden.

Durchführung

1. Das ist ganz simpel.
2. Gehen Sie bitte in den Ordner Information und aktivieren Sie die README.
   - Da der Systemeditor der README fest zugeordnet wurde, können Sie die Suchfunktion des Systemeditors nutzen. Suchen Sie als erstes nach dem Begriff:

     `Installation`
   - Suchen Sie nun nach dem Begriff

     `Kompatibilität`
   - und lesen Sie, was dazu ausgesagt wird.
3. Nun nehmen Sie als Suchbegriff:

   `Systemabschluß`
   - Blättern Sie ein wenig in dem Thema herum, bevor wir sofort weitergehen.
4. Jetzt suchen Sie bitte nach:

   `Arbeitsoberfläche`
   - und blättern Sie auch hier ein wenig herum.
5. Suchen Sie jetzt bitte nach:

   `Drucken`
6. und gleich danach nach:

   `Winsleuth Gold`
7. Jetzt suchen Sie bitte nach:

   `CorelDraw`

- In diesem Block sollten Sie sich ein wenig länger aufhalten. Suchen Sie z.B. nach dem Namen einer Ihnen bekannten Anwendung.
- Sie werden feststellen, daß hier eine Vielzahl von Anwendungen und Hardware behandelt werden, z.B:

  `SatisFAXtion`

  und

  `WordPerfect`

  nun etwas HW betreffend:

  `BILDSCHIRMADAPTER`

  und

  `AUDIOADAPTER`

Hierbei möchte ich es jetzt bewenden lassen. Denken Sie bei evtl. Unverträglichkeiten daran, als erstes die README zu Rate zu ziehen.

7.10.2 Übung: Betriebssystemstart bei Schwierigkeiten

Ziel der Übung

Sie erfahren, wie Sie bei so einem Problem sich durch einfachen Trick helfen können.

Durchführung

Wenn Ihr OS/2 aufgrund Problemen, die eine geöffnete Anwendung hervorruft, sich nicht sauber starten läßt, führen Sie bitte folgende Schritte durch.

1. Sie starten das System und warten solange bis der weiße Schirm und der Mauszeiger erscheinen.
2. Jetzt ist der Zeitpunkt gekommen. Drücken Sie die Tastenkombination links STRG + links SHIFT + F1 und zwar solange bis die ersten Symbole auf der Oberfläche erscheinen.
   - Wenn die Festplatte nicht angesteuert werden sollte, lassen Sie die Taste „F1“ kurzzeitig los.
3. Lassen Sie nun die Tasten los.

Was passiert bei diesem Vorgang?
Es wird einfach die eingebaute Funktion, beim Systemabschluß offene Objekte und Anwendungen mit zu starten/öffnen umgangen.

Sie haben auf diese Weise Ihre eigene Oberfläche aber eben ohne die Problem hervorrufenden Elemente aufgerufen.

Kapitel 8. Datenbestandsverwaltung

Ziele dieses Kapitels

Dieses Kapitel macht Sie mit den Möglichkeiten der Verwaltung von Daten und Programmen auf Festplatten und Disketten vertraut.

Referenzen

- Befehlsreferenz

Inhalt dieses Kapitels

8.1 Allgemeines 8-3
8.2 Plattenorganisation 8-5
  8.2.1 Was verbirgt sich hinter der FAT? 8-6
8.3 Das High Performance File System 8-7
  8.3.1 Was ist das HPFS, high performance file system? 8-7
  8.3.2 Übung: Erweiterte Attribute 8-11
  8.3.3 Echte Namen und Titel 8-12
  8.3.4 ÜBUNG: Lange Dateinamen bei HPFS und FAT 8-13
8.4 Anlegen und Löschen von Dateien 8-14
  8.4.1 Anlegen von Dateien mit Schablonen 8-14
  8.4.2 Anlegen von Dateien aus Programmen heraus . 8-15
  8.4.3 Löschen von Datenbeständen 8-16
  8.4.4 Wiederbelebung 8-17
  8.4.5 ÜBUNG: Wiederbelebung gelöschter Dateien . . 8-19
  8.4.6 Übung: Ergänzung zum Thema "UNDELETE" . . . 8-21
8.5 Sichern von Datenbeständen 8-22

8.1 Allgemeines

Das Objekt-orientierte Konzept des OS/2 Version 2 erfordert auch bei der Pflege von Datenbeständen eine neue Vorgehensweise.

Ein ganz herausragendes Merkmal der Datenstrukturen ist die Einbindung der Arbeitsoberfläche in die Plattenstrukturen: Ein Ordner am Bildschirm ist **auch** gleichzeitig ein Verzeichnis auf einer Festplatte oder Diskette.

Es gibt im OS/2 Version 2 keinen Datei-Manager mehr. Daher sind alle Aufgaben aus den Ordnern heraus zu erledigen. Das ähnelt der Sicht des Datei-Manager, nur sind die „Fenster“ mit den Inhalten nicht mehr in einem Programm zusammengefaßt. Die Funktionen des Datei-Manager sind durch Funktionen im OS/2 Version 2 abgedeckt. Daher ist ein Datei-Manager auch gar nicht erforderlich.

Sollte ein Benutzer jedoch Wert auf eine solche Anordnung legen, so läßt sich das eleganter als unter anderen Betriebssystemen verwirklichen. Dazu wird folgendes gemacht:

1. Arbeitsordner anlegen mit Namen z.B. Datei-Verwaltung,
2. Referenz-Kopien aller gewünschten Laufwerke und der gewünschten Ordner (Verzeichnisse) in diesem Ordner anlegen,
3. Nach Öffnen des jeweiligen Ordners können alle Dateioperationen durchgeführt werden.

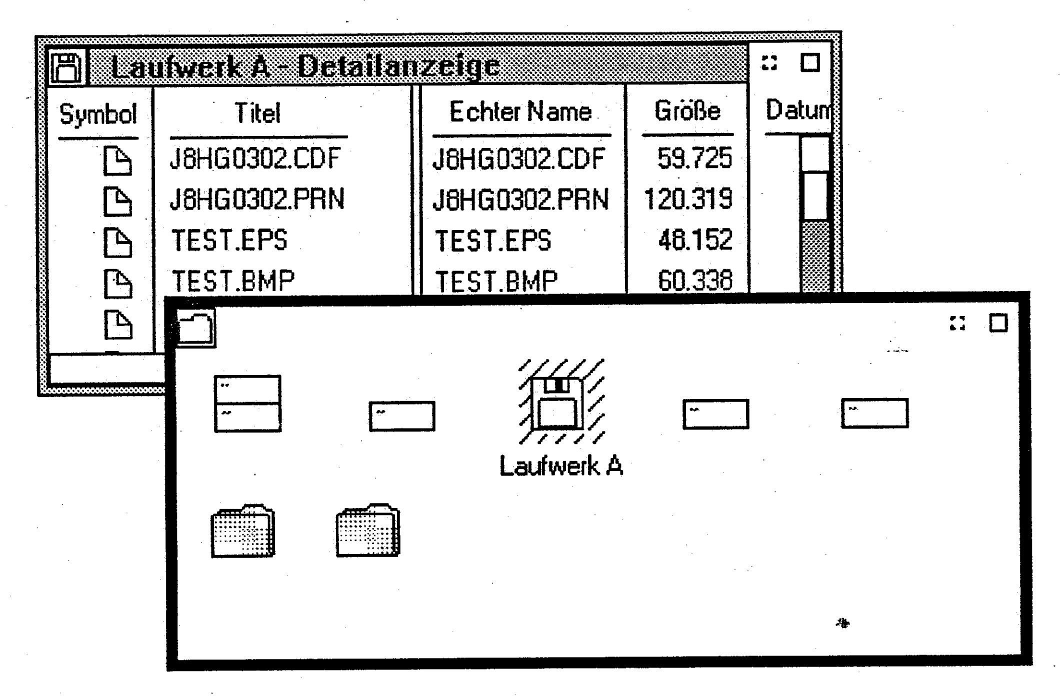


Abbildung 8-1. Beispiel eines Ordners "Dateiverwaltung" (J8HG0701)

Ein Vorteil dieser Anordnung ist die Funktion der Arbeitsordner: Nach Öffnen des Arbeitsordner werden die zuletzt offenen Ordner gleich wieder angezeigt, ein anderer Vorteil ist die Möglichkeit, Dinge für einen Endbenutzer so anzulegen, daß dieser nur sieht, was er bearbeiten soll und darf, beispielsweise Ordner mit bestimmten Hilfsprogrammen oder mit bestimmten Daten.

8.2 Plattenorganisation

Bei der Plattenorganisation wurde sowohl die Kompatibilität zu bisherigen Betriebssystemen wie auch die Möglichkeiten zukünftiger Plattendateisysteme berücksichtigt. Daher wurde eine Struktur geschaffen, welche sowohl einfache Benutzung wie auch beste Ausnutzung der System-Ressourcen sicherstellt.

Was für den Endbenutzer einfach ist, kann für den Administrator zunächst eine größere Komplexität mit sich bringen. Diese ist jedoch logisch und verständlich.

Um dem Anwender eine leichte Umstellung der Daten von DOS nach OS/2 zu erlauben und um die Koexistenz von DOS und OS/2 zu sichern, wurde die traditionelle Speicherorganisation auf **Disketten** erhalten. Diese Art der Datenspeicherung wird kurz **FAT** - *File Allocation Table* - genannt.

Auf dem Datenträger werden gespeichert:

- Datenträgernamen 11 Stellen
- Universelle Datenträgeridentifikation, wird beim Formatieren automatisch vergeben und soll sich von jedem anderen Datenträger unterscheiden
- Systemdateien, wenn es sich um eine DOS-Diskette mit Urlader handelt
- Verzeichnisse
- Dateinamen (8.3 Stellen) und Dateiattribute ARHS

8.2.1 Was verbirgt sich hinter der FAT?

Die FAT war das erste für Personal Computer konzipierte Plattenorganisationssystem. Es wurde primär für Disketten entwickelt. Einige Charakteristika sind:

- Die FAT steht auf der äußersten Spur der Platte.
- Je größer die Platte, desto größer ist auch die FAT.
- FAT ist die Bezeichnung des Organisationssystems, aber auch gleichzeitig der Name der notwendigen Zugriffstabelle, der *file allocation table* auf dem Datenträger.
- Die *file allocation table* wird vom Betriebssystem in der Regel doppelt auf dem Datenträger zur Verfügung gestellt. Im Falle, der Zerstörung der 1. FAT wird mittels CHKDSK die 1. durch die 2. ersetzt.
- Eine FAT-formatierte Platte ist organisiert:
  - *Boot sector*
  - *file allocation table* und gleich dahinter das Doppel
  - Hauptverzeichnis
  - mit Daten beschreibbare Cluster
- Die FAT enthält die Adressen aller Cluster.
- Die Verzeichnisstruktur ist eine unsortierte lineare Liste, sie muß daher von vorne bis hinten durchsucht werden.
- Wenn ein Cluster frei ist, wird das durch den bit-Stand 0 vermerkt.
- Pro Cluster werden 16 - 24 bit-Stellen benötigt.
- Ein Cluster kann abhängig von der DOS-Version und der Datenträgergröße zwischen 1 und 64 Sektoren mit je einer Kapazität von 512 Bytes beinhalten.

OS/2 ermöglicht die FAT-Adressierung bis zu 2 GB pro logischen Datenträger. Die sinnvolle Grenze liegt allerdings bei ca. 60 MB. Der FAT-Cache (*diskcache*) kann theoretisch bis zu 64 MB groß definiert werden.

8.3 Das High Performance File System

8.3.1 Was ist das HPFS, high performance file system?

Das **HPFS** wurde entwickelt um den bei mittlerweile großen Datenräumen sich verschlechternden Durchsatz zu erhöhen. Es wurde ein Organisationssystem entwickelt, daß sich von den vorgegebenen Strukturen frei machte mit der Maßgabe:

- Durchsatz
- Sicherheit
- Organisation
- Platzeinsparung
- Lange Dateinamen
- weitere Attribute

zu verbessern, resp. zu ermöglichen. Hier einige wichtige Charakteristika des neuen Systems:

- Belegungseinheit ist der Sektor, entsprechend 512 bytes.
- 1 bit (1=frei) sagt aus, ob der Sektor belegt oder frei ist.
- Die Belegungsinformation liegt in der Nähe der Sektoren.
- Die Sektoren sind zusammengefaßt zu zusammenhängenden Streifen von 8 bzw. 16 MB.
- Der Verzeichnisstreifen befindet sich in der Mitte der logischen Platte.
- Die Verzeichnisstruktur wird in Form eines *balanced B-tree* erstellt und gewartet.
- Vor dem Schreiben der Datei auf den Datenträger wird erst die Größe ermittelt und dann nach entsprechend großen zusammenhägenden Streifen gesucht.
- Der Verzeichnisstreifen enthält nicht nur die Information wo die gesuchte Datei beginnt, sondern auch die Anzahl der belegten Sektoren.
- Intelligentes caching.
- Pufferspeicher für Daten und Verzeichnisse
- Dateinamen sind auf der Platte sortiert

- Das Schreiben von Daten kann über Puffer mit maximaler Wartezeit erfolgen
- Aufzeichnen des letzten Zugriffes auf Dateien
- ca. 400 KB Treiber und Arbeitsbereich resident im Hauptspeicher
- Nach ursprünglichem Programmladen kann Platte automatisch auf Integrität geprüft werden
- Unterstützt bis zu 16 Bereiche *(partitions)* je 2199 GB, empfohlen wird allerdings die Partitionen nicht größer als 512 GB zu definieren.
- FAT-Kompatibel auf der API-Ebene
- Neue API-Funktionen zum Erstellen und Benutzen des HPFS
- Zugriff sowohl von OS/2-Sitzungen aus wie auch aus einer VDM heraus unterstützt (VDM auf DOS-übliche Attribute beschränkt)
- Der Zugriff von Original DOS-Versionen ist unter Verwendung von FSFILTER.SYS möglich.
- Neuer Eintrag in CONFIG.SYS:
IFS=C:\OS2\HPFS.IFS /AUTOCHECK
- Neuer Parameter für das Formatieren:
FORMAT X: /FS:HPFS.
- HPFS trägt die letzte Benutzung einer Datei in die EA ein
- HPFS erlaubt Namen mit bis zu 254 Stellen. Diese dürfen Sonderzeichen enthalten. Nicht erlaubt sind:

  / \ | < > ? * " & –

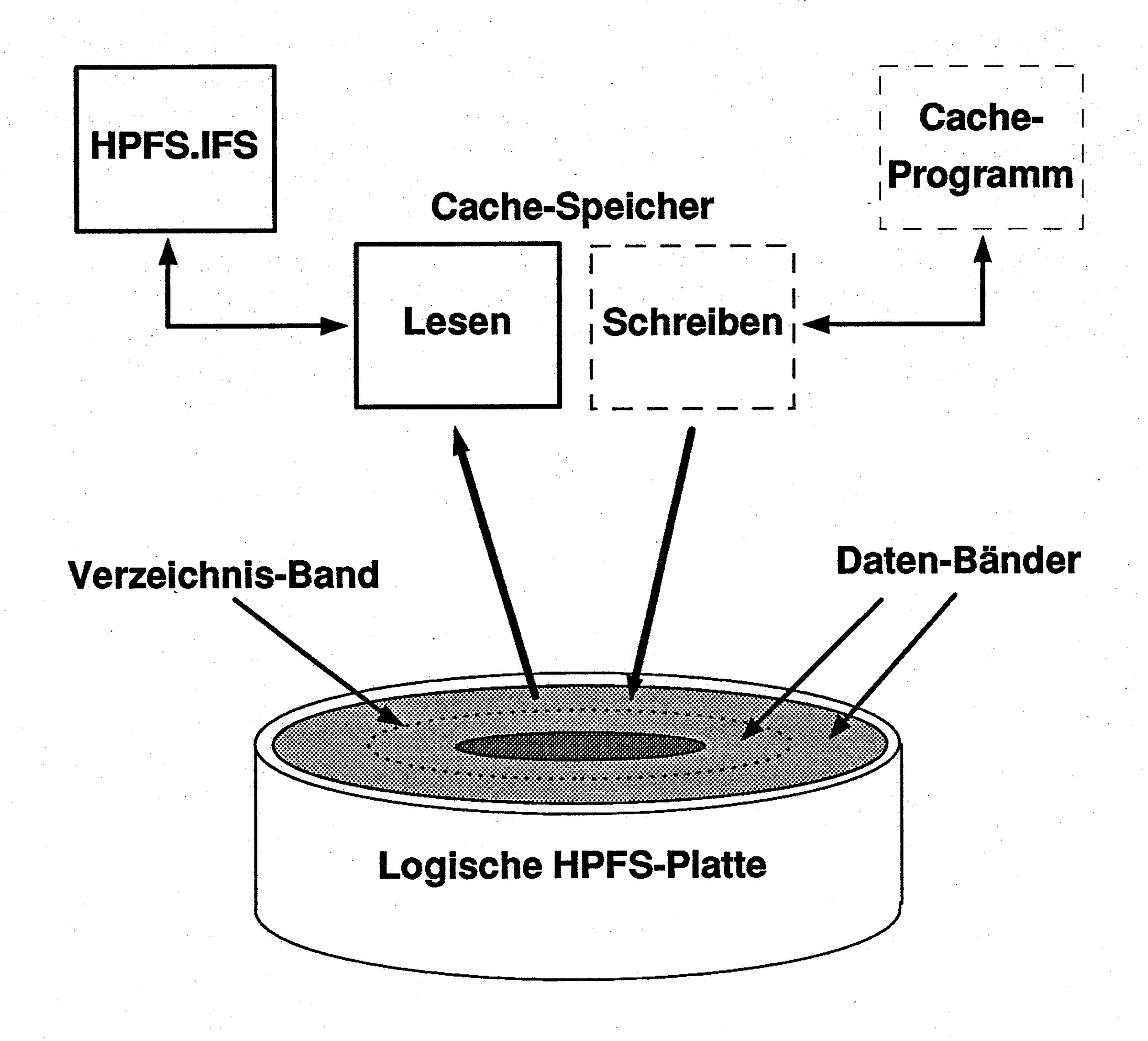


Abbildung 8-2. Konzept des HPFS (J81GHPFS)

Festplatten können nach beiden Systemen organisiert werden. Zusätzlich zu den traditionellen Attributen des FAT-Systemes sind unter HPFS folgende **Erweiterte Attribute** gekommen:

- Kurzreferenz
- Schlüsselwörter
- Kommentare
- Datentyp
- Protokoll
- Zuordnung von Piktogrammen

Die Attribute werden mit den Einstellungen Seite 2 und 3 zugewiesen.

Die Erweiterten Attribute werden in Dateien mit dem Namen **EA DATA. SF** gespeichert. Es handelt sich um „versteckte" Dateien, welche einmal je logischem FAT-Laufwerk vorhanden sind. Beim Kopieren oder Verschieben eines Programmes oder einer Datei werden diese Attribut-Dateien entsprechend geändert.

Beim Kopieren auf Diskette wird dort eine neue Datei für die Attribute angelegt. Werden die Attribute nicht mehr benötigt, so können sie mit EAUTIL abgetrennt werden. Das kann bei der Übernahme von Daten mit Diskette zu einer DOS-Maschine der Fall sein. Kommen veränderte Daten auf der Diskette wieder zurück, so können die Attribute mit EAUTIL wieder zugeordnet werden.

Die Erweiterten Attribute können bis zu 64 KB je Datei belegen. Normal sind jedoch einige ..zig Bytes.

Merke

Gelegentlich werden Logische Zusammengehörigkeiten von Dateien als Erweiterte Attribute gespeichert. Diese Verknüpfungen werden von entsprechenden Programmen (z.B. Installationsprozedur) dazu verwendet, alle zusammengehörigen Programme zusammen zu verwenden.

8.3.2 Übung: Erweiterte Attribute

Ziel der Übung

Sie fügen einer Datei Erweiterte Attribute zu und erfahren, wie Sie diese bei Bedarf abtrennen können.

Durchführung

1. Beginnen Sie bitte, indem Sie, sofern noch nicht vorhanden, die autoexec.bat als butoexec.bat kopieren.
2. Rufen Sie das Kontextmenü dieser Datei auf.
3. Die Notizbuchseite Datei hat zwei weitere Seiten. Blättern Sie bitte weiter auf die 3. dieser Seiten und tragen Sie einen Kommentar und ein Schlüsselwort ein.
4. Schließen Sie das Notizbuch wieder.
5. Jetzt öffnen Sie eine OS/2-Befehlszeile und geben ein

   ```
   eautil butoexec.bat butoexec.atr /s
   ```
6. Sehen Sie sich nun die Dateinamen mithilfe des Directory-Aufrufs an.
7. Was stellen Sie fest?
   - ____
8. So, jetzt werden wir den Prozeß umdrehen. Geben Sie bitte ein:

   ```
   eautil butoexec.bat butoexec.atr /j
   ```
9. Sehen Sie sich nun die Dateinamen mithilfe des Directory-Aufrufs an.
10. Was stellen Sie fest?
    - ____

8.3.3 Echte Namen und Titel

Um die früheren, teils kryptischen Namen und mehrdeutigen Namenserweiterungen zu ersetzen, wurden die Datei-Namen in zwei Namen aufgeteilt:

1. **Titel**, das sind Namen am Bildschirm.
2. **Reale Namen**, das sind die Namen auf den Datenträgern.

Die Namenserweiterungen dienten oft als Kennzeichnung des Inhaltes, stellten also einen Dateityp dar. Dieser ist im OS/2 Version 2 nur noch als eines der erweiterten Attribute festgelegt.

Weiterhin sollte es möglich sein, daß der Endbenutzer eine Datei am Bildschirm beliebig benennt, daß aber aus programmtechnischen Gründen der Name auf dem Datenträger unverändert bleibt. So ist es nun möglich, Titel zu vergeben, welche 2 oder mehr Zeilen am Bildschirm belegen.

Da es (zumindest in der nächsten Zeit) Programme gibt, welche sich noch nicht an den neuen Gegebenheiten orientieren, kann es gelegentlich zu Besonderheiten kommen, welche sorgfältig bearbeitet werden müssen. Das gilt insbesondere für die Verwendung von Dateierweiterungen alter Art anstelle der Erweiterten Attribute. Daher wird beim Vergeben eines Titels für eine Datei vom OS/2 ein Warnung ausgegeben, welche diesen Sachverhalt anspricht.

8.3.4 ÜBUNG: Lange Dateinamen bei HPFS und FAT

Ziel der Übung

Sie erstellen Dateien mit langen Namen und stellen dann fest, wie OS/2 damit umgeht.

Durchführung

1. Beginnen Sie bitte, indem Sie aus dem Objekt Laufwerke heraus mit Hilfe der STRG-Taste und der Maustaste 2 die DOS-Autoexec.bat unter den Namen

   ```
   autoexec1.bat
   autoexec2.bat
   autoexecvers3.bat
   autoexecvers4.bata
   autoexecvers.5.bata
   ```

   auf eine FAT-Platte oder Diskette kopieren.

2. Sehen Sie sich jetzt die Dateien mit Hilfe des Objektes Ihres entsprechenden Laufwerkes an und zwar mit der Einstellung: Detailanzeige.
3. Was stellen Sie fest?
   - ______________________________
4. Öffnen Sie nunmehr ein OS/2-Fenster und rufen Sie das directory auf.
5. Was stellen Sie fest?
   - ______________________________
6. Jetzt öffnen Sie ein DOS-Fenster und rufen das directory auf.
7. Was stellen Sie fest?
   - ______________________________

Sie sehen, es gibt auch mit den langen Namen kein Problem, selbst wenn Sie die Dateien auf eine Diskette kopieren.

8.4 Anlegen und Löschen von Dateien

Je nach Art der verwendeten Programme sind verschieden Möglichkeiten zum Anlegen von Dateien möglich oder auch notwendig:

1. **Traditonelle Art** : Sichern einer Datei von einem Programm aus, d.h. über eine Menüauswahl wie Sichern unter... .
2. **OS/2 Version 2-Art** : Benutzung von Schablonen, d.h. von einer Schablone wird eine Kopie abgezogen und diese dann als Datenobjekt verwendet.

8.4.1 Anlegen von Dateien mit Schablonen

Wird eine der Standard-Schablonen von OS/2 Version 2 benutzt, so hat diese weder Inhalt noch einen geeigneten Namen, da ja ein Standard-Namen verwendet wird. Dieser ist am Bildschirm leicht zu überschreiben.

Diese Methode ist für Endbenutzer jedoch nicht sonderlich praktisch. Daher sollten unbedingt Schablonen eingerichtet werden, welche schon vorgegebene Inhalte haben und auch durch Anklicken sofort das richtige Programm aufrufen.

Einige Programme legen von sich aus Schablonen im Ordner Schablonen an. Es kann sinnvoll sein, diese Schablonen in einen entsprechenden Arbeitsordner zu legen.

8.4.2 Anlegen von Dateien aus Programmen heraus

Werden Programme direkt aufgerufen, so lassen sich meist unter dem Menüpunkt Sichern als... Dateien an beliebiger Stelle erzeugen. Das geht selbstverständlich auch unter OS/2 Version 2. Die Besonderheit ist die Struktur der Arbeitsordner, wie sie in Abb. 8-3 zu sehen ist.

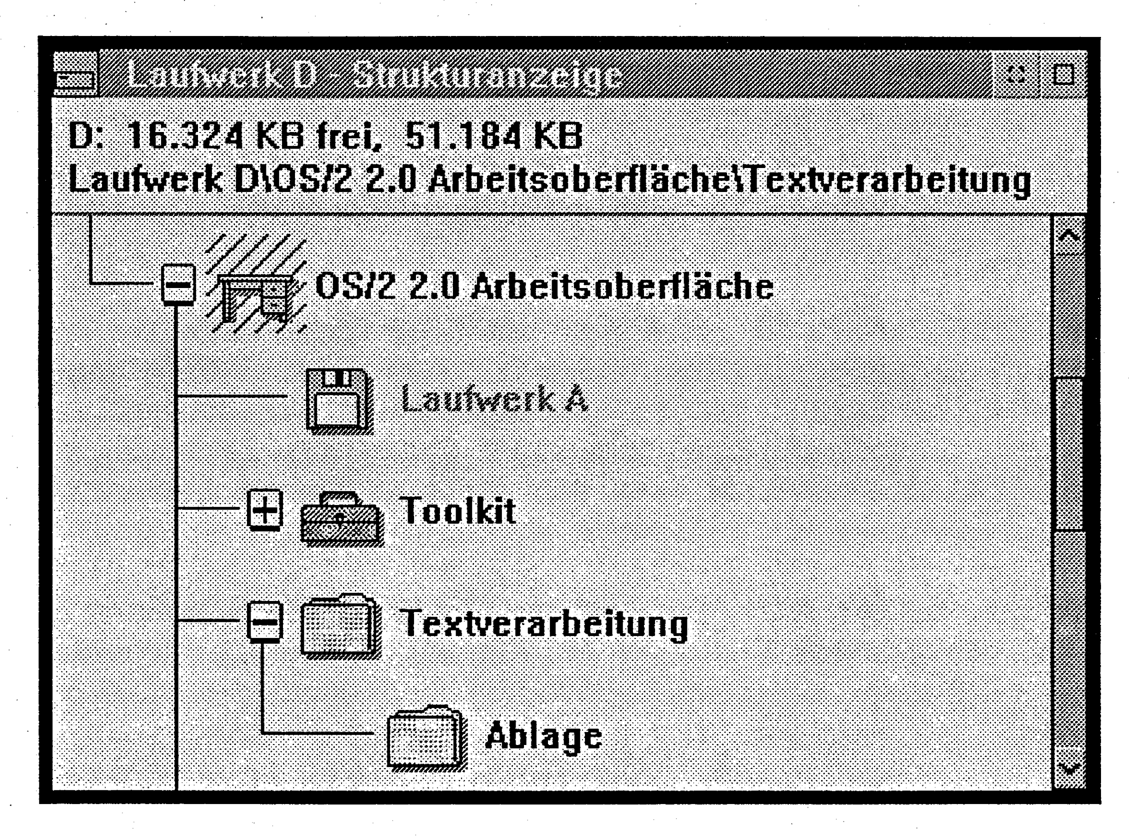


Abbildung 8-3. Arbeitsoberfläche-Struktur ist eingebunden in Plattenstruktur (J8HG0303)

8.4.3 Löschen von Datenbeständen

OS/2 Version 2 bietet eine Einrichtung an, welche das Risiko beim Löschen von Datenbeständen verringert. Information über gelöschte Dateien wird an einem von Ihnen zu bestimmenden Platz gespeichert. Das erlaubt ein Wiederherstellen der gelöschten Datei(en).

In der Datei CONFIG.SYS ist ein Eintrag, welcher das Anlegen, von Ordnern (Verzeichnissen) mit dem Namen Delete bewirkt.

```
SET DELDIR=C:\DELETE,512;D:\...
```

Ein solcher Ordner kann einmal je logischem Laufwerk angelegt werden. Die Zahl nach dem Namen des Ordners gibt die Größe in KB an.

Nach der Installation von OS/2 Version 2 ist der Befehl durch ein REM unwirksam. Wenn Sie nun dieses REM entfernen und das System danach abschließen und neu starten, werden Sie den (die) Ordner nicht sofort finden. Erst nach dem ersten Löschen einer Datei auf dem entsprechenden Laufwerk ist das Verzeichnis vorhanden.

Die Werte für Pfad und Maximalgröße auf jedem logischen Laufwerks sind durch ein Komma voneinander getrennt; die Angaben zu logischen Laufwerken sind durch Semikolons voneinander getrennt. Wird in einer beliebigen Sitzung eine Datei mit dem Befehl DEL oder ERASE gelöscht, so wird diese Datei in das Verzeichnis kopiert, das in der Anweisung DELDIR für das betreffende logische Laufwerk angegeben ist.

Wenn die Anzahl gelöschter Dateien die maximale Größe des angegebenen Verzeichnisses überschreitet, werden automatisch Dateien aus diesem gelöscht, so daß die angegebene Maximalgröße nicht überschritten wird. Beim Löschen wird die Reihenfolge FIFO (first in, first out) verwendet. Sollen gelöschte Dateien nicht in einem Verzeichnis gespeichert werden, so muß der Verzeichnisname aus der Anweisung DELDIR gelöscht werden.

8.4.4 Wiederbelebung

Mit dem Befehl UNDELETE können gelöschte Dateien wiederhergestellt werden. Dies geschieht dadurch, daß das endgültige Löschen der Dateien verzögert wird, so daß zuvor noch auf sie zugegriffen werden kann. Voraussetzung ist die Einrichtung entsprechend 8.4.3, „Löschen von Datenbeständen“ auf Seite 8-16 .

Der Befehl UNDELETE hat folgende Syntax:

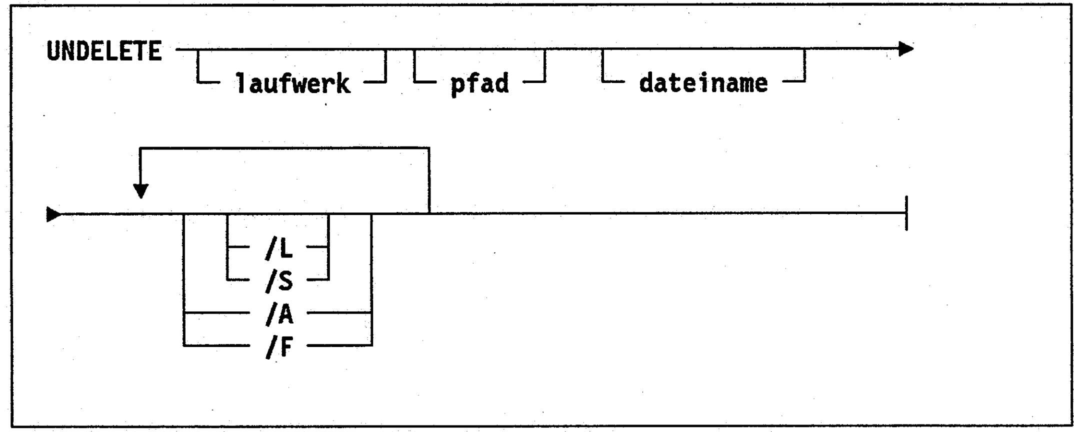


Abbildung 8-4. Syntaxübersicht UNDELETE (J8HTUDEL)

Wird der Befehl UNDELETE angegeben, wird die Datei (falls noch verfügbar) aus dem verdeckten Verzeichnis abgerufen und wieder in den ursprünglichen Pfad kopiert. Würde durch die Wiederherstellung eine bestehende Datei überschrieben werden, wird eine entsprechende Bestätigungsnachricht angezeigt; gibt der Benutzer nicht an, daß die Datei überschrieben werden soll, so wird der Vorgang ignoriert.

UNDELETE kann sowohl in DOS-Sitzungen als auch in OS/2-Sitzungen verwendet werden. Der Platz, den die Dateien, die wiederhergestellt werden können, auf dem Datenträger beanspruchen, wird als belegter Speicherplatz angezeigt.

Das folgende Beispiel zeigt die Benutzung von UNDELETE in einem OS/2-Fenster.

```
D:\]undelete /l /s
D:\* wird wiederhergestellt.

D:\OS2\BITMAP\ABC.SCR
D:\OS!2 2.0 ARBEITSOBERFLäCHE\TEXTVERARBEITUNG\MUSTER!1
D:\CMDREF.CP
D:\CMDREF.CP
D:\SPOOL\CONFIG.BA2

[D:\]
```

Abbildung 8-5. Anzeige wiederherstellbarer Dateien

Achtung!

Das Ausrufungszeichen im Dateinamen ist gültig und richtig!

8.4.5 ÜBUNG: Wiederbelebung gelöschter Dateien

Ziel der Übung

Sie lernen die erforderlichen Schritte kennen, löschen eine Datei und holen sich die diese mit dem Dienstprogramm undelete wieder.

Durchführung

Bevor wir mit der eigentlichen Übung beginnen können, müssen wir erst die Voraussetzungen dafür schaffen.

1. Sofern noch nicht geschehen, müssen wir die CONFIG.SYS editieren. Öffnen Sie die Datei CONFIG.SYS und nehmen Sie dem Statement

   ```
   REM SET DELDIR=C:\DELETE,512.........
   ```

   den remark- (=Bemerkungs-) Status.

2. Sichern Sie die Datei, führen Sie einen Systemabschluß durch und booten Sie erneut.

3. Nachdem Ihr System wieder arbeitsfähig ist, löschen Sie die Datei „butoexec.bat", wobei es Ihnen überlassen ist, auf welche Weise Sie das tun. Also entweder

   mit Hilfe des Kontextmenüs
   oder
   aus einer Befehlszeile heraus
   oder
   indem Sie die Datei per drag and drop über den Papierkorb ziehen.

4. Jetzt wollen wir die Datei wiederholen. Geben Sie bitte in einer OS/2-Befehlszeile ein:

   ```
   undelete /l
   ```

5. Nun geben Sie ein

   ```
   undelete /s
   ```

6. Anschließend prüfen Sie, ob die „butoexec.bat" wieder zur Verfügung steht.

   Hat es geklappt?

7. Wenn Sie eine Datei nun auch physisch aus dem „DELETE"-Verzeichnis löschen möchten, geben Sie bitte ein:

   ```
   undelete /f
   ```

   oder, wie es in unserem Fall zuträfe

   ```
   undelete butoexec.bat /f
   ```

8.4.6 Übung: Ergänzung zum Thema "UNDELETE"

Ziel der Übung

Sie machen sich mit den möglichen Parametern für undelete vertraut.

Durchführung

1. Finden Sie die heraus, wie Sie mit undelete alle gelöschten Dateien eines Laufwerkes wieder aktivieren können, ohne daß Sie für jede einzelne Datei bestätigen müssen.
2. Schreiben Sie hier die Schritte auf:
   - ______________________
   - ______________________
   - ______________________

8.5 Sichern von Datenbeständen

Um Dateien mit vorhandener Hardware und Software sichern zu können, gibt es den Befehl BACKUP.

Die Syntax ist wie folgt:

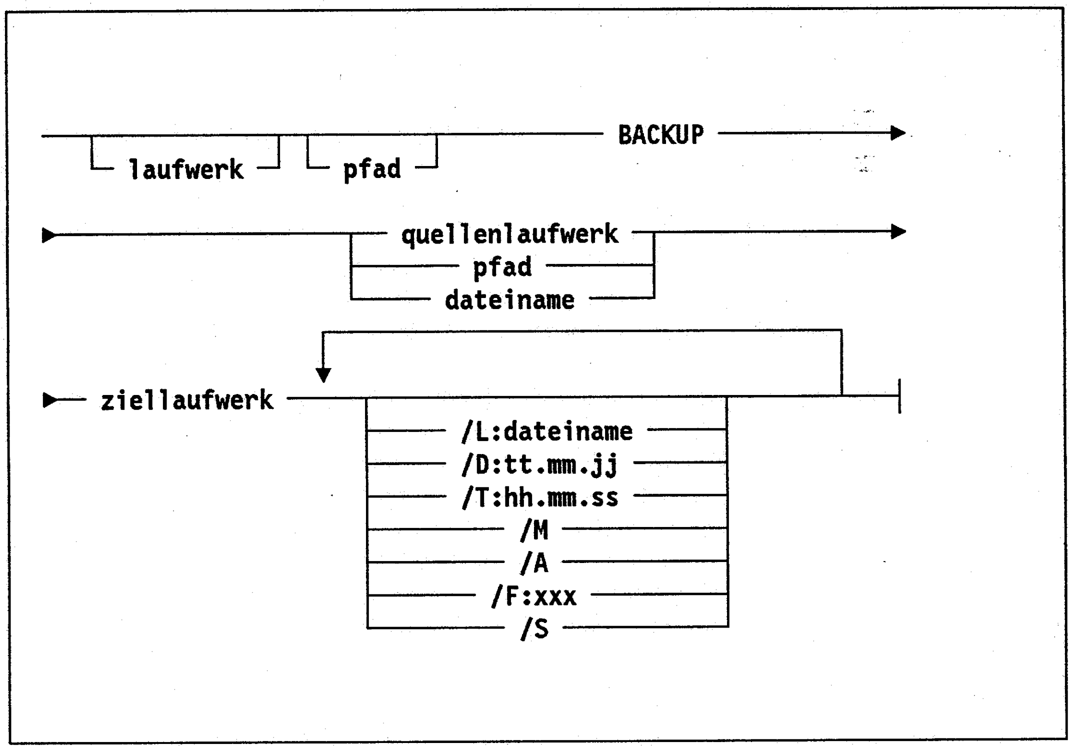


Abbildung 8-6. Syntaxübersicht BACKUP (J8HTBACK)

Wichtig!

1. Die Systemdateien COMMAND.COM und CMD.EXE, verdeckte Systemdateien und geöffnete Dateien mit Dynamischen Link-Bibliotheken (.DLL) können mit BACKUP nicht gesichert werden. Auch Dateien, die mit „Deny Read/Write" geöffnet wurden, können mit BACKUP nicht gesichert werden. Wird versucht, eine dieser Dateien zu sichern, wird eine Systemnachricht ausgegeben.

2. Besteht gemeinsamer Zugriff auf Dateien, kann der Benutzer nur jeweils die Dateien sichern, für die er eine Zugriffsberechtigung hat. Versucht der Benutzer, auf eine Datei zuzugreifen, für die er keine Zugriffsberechtigung hat, wird von BACKUP eine Nachricht angezeigt, die besagt, daß die betreffende Datei nicht gesichert werden konnte.

3. Die gesicherten Dateien stehen nicht im Stammverzeichnis der Zieldiskette. Im Stammverzeichnis der Zieldiskette werden zwei Dateien erstellt: BACKUP.XXX und CONTROL.XXX. Die Datei BACKUP.XXX enthält die gesicherten Dateien, in der Datei CONTROL.XXX sind die Pfade, die Dateinamen und andere Steuerinformationen gespeichert.

4. Erweiterte Attribute: Mit dem Befehl BACKUP werden auch die zu einer Datei oder einem Verzeichnis gehörenden erweiterten Attribute gesichert. Ein erweitertes Attribut ist ein spezieller Bereich, in dem Daten gespeichert werden, die eine Beschreibung der Datei bzw. des Verzeichnisses enthalten. Diese Beschreibung wird vom Betriebssystem und von Anwendungsprogrammen verwendet. Sollen Dateien oder Verzeichnisse mit erweiterten Attributen gesichert werden, muß OS/2, Version 1.2 oder eine spätere Version, verwendet werden, damit alle erweiterten Dateiattribute gesichert werden.

Es ist erkennbar, daß vielfältige Steuerungsmöglichkeiten ein gezieltes Sichern nur der veränderten Dateien erlaubt. Durch gute Planung und Organisation ist auch mit Disketten eine vollständige Datensicherung durch Fortschreibung erreichbar.

Mit RESTORE können eventuell versehentlich gelöschte Dateien wiederhergestellt werden:

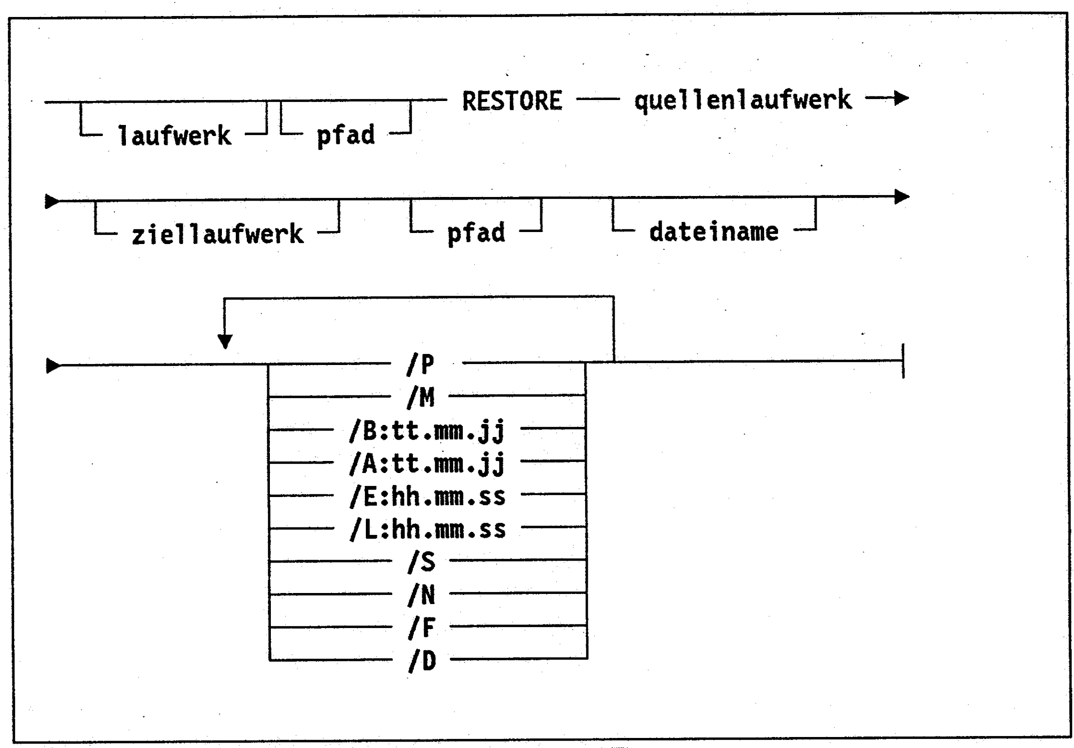


Abbildung 8-7. Syntaxübersicht RESTORE (J8HTREST)

Wichtig!

Erweiterte Attribute: Während der Ausführung des Befehls RESTORE werden die erweiterten Attribute einer mit BACKUP gesicherten Quellendatei bzw. eines Quellenverzeichnisses auf den Zieldatenträger kopiert. (Siehe den Parameter /F.) Als erweiterte Attribute bezeichnet man einen speziellen Bereich, in dem Daten gespeichert werden, die eine Beschreibung der Datei bzw. des Verzeichnisses enthalten, die vom System oder von Anwendungsprogrammen benötigt wird.

Sollen Dateien mit erweiterten Attributen zurückgeschrieben werden, muß OS/2, Version 2 verwendet werden, um sicherzustellen, daß alle erweiterten Datei- oder Verzeichnisattribute wiederhergestellt werden. Die erweiterten Attribute eines gesicherten Quellenverzeichnisses werden nur auf den Zieldatenträger kopiert, wenn das entsprechende Unterverzeichnis auf dem Zieldatenträger noch nicht vorhanden ist.

Kapitel 9. Startmöglichkeiten

Ziele dieses Kapitels

In diesem Kapitel lernen Sie die Möglichkeiten, Anwendungen beim Systemstart mitzustarten und die Bedeutung von Stapeldateien und deren Aufgaben beim Systemanlauf kennen.

Inhalt dieses Kapitels

9.1 Allgemeines zu Stapeldateien 9-3
9.1.1 Einsatz von Stapeldateien 9-3
9.1.2 DOS- und OS/2 Stapeldateien 9-4
9.1.3 Befehle für Stapeldateien 9-6
9.2 Systemstart (STARTUP) 9-8
9.2.1 STARTUP-Datei 9-9
9.2.2 STARTUP-Ordner 9-12

9.1 Allgemeines zu Stapeldateien

Eine Stapeldatei ist eine Datei, die einen oder mehrere Befehle enthält, die von OS/2 nacheinander gestartet werden. Stapeldateien können sowohl für DOS- als auch für OS/2-Sitzungen erstellt werden. Beim Ausführen einer Stapeldatei wird der „Befehlsstapel" zeilenweise vom OS/2-Befehlsprozessor verarbeitet. Bestimmte immer wiederkehrende Befehlsfolgen können in eine Stapeldatei aufgenommen, also „gestapelt" werden.

9.1.1 Einsatz von Stapeldateien

Stapeldateien werden demnach verwendet, wenn stattdessen eine Reihe von einzelnen Befehlen nacheinander eingegeben werden müßte. Durch ihren Einsatz läßt sich der Zeitaufwand sowie die Fehlerquote bei der Eingabe erheblich reduzieren.

Beispiel für gestapelte Anforderungen

Wenn Sie eine Anwendung, bespielsweise ein Adreßverwaltungsprogramm mit dem Namen ADRESSE, aufrufen, vorher evtl. das Systemdatum aktualisieren wollen, lassen sich folgende Anforderungen auflisten:

1. Ändern Systemdatum.
2. Wechseln in das Verzeichnis:

   `D:\ANWEND\ADRESSE`
3. Aufrufen des Programms:

   `ADRESSE.EXE`

Anwendungsbeispiel als Stapeldatei

Für die aufgelisteten Anforderungen können Sie eine Stapeldatei ADR.CMD die folgende Befehle beinhaltet, erstellen:

```
DATE
D:
CD D:\ANWEND\ADRESSE
ADRESSE.EXE
```

Aufruf der Stapeldatei

Zur Erstellung der Stapeldatei verwenden Sie einen Texteditor und geben der Datei einen kurzen aber sprechenden Namen. Für das obige Anwendungsbeispiel würde sich „ADR“ anbieten. Statt der ursprünglich 3 Befehle, geben Sie nur noch „ADR“ in die Befehlszeile ein und nach Angabe des Datums wird Ihr Adreßverwaltungsprogramm aufgerufen.
Berücksichtigen Sie bei der Namensgebung, daß Sie keine Namen interner Systembefehle wie z.B. COPY oder DIR verwenden.

9.1.2 DOS- und OS/2 Stapeldateien

Stapeldateien, auch *Batch Files* genannt, haben die Dateierweiterung **CMD**, wenn sie unter OS/2 laufen und die Erweiterung **BAT**, wenn sie unter DOS laufen. Für DOS erstellte Stapeldateien (.BAT), können von einer OS/2 Sitzung aus gestartet werden. Der OS/2 Befehlsprozessor erkennt die BAT Dateierweiterung, ruft automatisch den DOS Befehlsprozessor auf und beendet diesen, wenn die Stapelverarbeitung abgeschlossen ist.
Sie können DOS-Stapeldateien auch direkt unter OS/2 laufen lassen, indem Sie die Dateiweiterung durch CMD ersetzen. Die Datei darf sich aber nicht auf ein, nur unter DOS laufendes, Programm beziehen.

CMD-Dateien unter OS/2

In der Basisversion von OS/2 finden Sie folgende CMD-Dateien auf Ihrem Systemlaufwerk:

```
D:\OS2\HELP.CMD
D:\OS2\REXXTRY.CMD
```

Die HELP-Stapeldatei dient dem Aufruf von Hilfefunktionen, wie die Funktionsbeschreibung eines OS/2-Befehls oder einer Fehlermeldung, bzw. dem Ausschalten der Hilfszeile innerhalb einer OS/2-Sitzung heraus.

Hinweis

Auch Stapeldateien mit REXX-Inhalt erhalten die Endung CMD!

BAT-Dateien unter OS/2

Die Stapeldateien unter OS/2 mit der Endung .BAT werden innerhalb der virtuellen DOS-Sitzungen benutzt.

```
D:\OS2\MDOS\HELP.BAT
D:\AUTOEXEC.BAT
```

Die HELP-Stapeldatei dient wie oben erwähnt der Hilfefunktion, in Fall innerhalb von DOS-Sitzungen.
Die AUTOEXEC.BAT ist die Datei, welche das DOS beim Systemstart ausführt. Innerhalb des OS/2 ist eine solche Datei nicht notwendig, Parameter für den Systemstart können aber in einer Systemstart-Stapeldatei mit dem Namen STARTUP.CMD festgelegt werden. Auf diese Möglichkeit wird am Ende dieses Kapitels eingegangen.

9.1.3 Befehle für Stapeldateien

Es stehen eine Reihe von Befehlen für Stapeldateien zur Verfügung, mit denen auch umfangreichere Anforderungen erfüllt werden. Zur Erstellung einer komplexeren Stapeldatei sind Programmierkenntnisse sicher von Vorteil.

Tabelle mit Befehlen für Stapeldateien

Die folgenden Befehle gelten für Stapeldatei- und Befehlsverarbeitungsoperationen:

Befehl	Bedeutung
CALL	Verschachtelung einer Stapeldatei innerhalb einer anderen Stapeldatei.
CMD	Starten eines weiteren OS/2-Befehlsprozessors.
COMMAND	Starten eines weiteren DOS-Befehlsprozessors.
ECHO	Aktivierung bzw. Inaktivierung der Anzeige von OS/2-Befehlen, während eine Stapeldatei ausgeführt wird.
ENDLOCAL	Erneute Aktivierung des Laufwerks, Verzeichnisses und der Variablen, die gültig waren, bevor der Befehl SETLOCAL ausgeführt wurde.
EXIT	Beenden des aktuellen Befehlsprozessors und Rückkehr zum vorigen Befehlsprozessor (falls vorhanden).
EXTPROC	Definition eines externen Prozessors für die Stapelverarbeitung.
FOR	Wiederholte Verarbeitung von Befehlen innerhalb einer Stapeldatei.
GOTO	Übergabe der Steuerung der Stapelverarbeitung an ein bestimmtes Sprungziel.
IF	Konditionelle Verarbeitung von Befehlen innerhalb einer Stapeldatei.
PAUSE	Unterbrechung der Verarbeitung einer Stapeldatei

PROTSHELL	Angabe der Benutzerschnittstelle und des OS/2-Befehlsprozessors.
REM	Anzeige von Kommentaren, die sich in einer Stapeldatei befinden.
SETLOCAL	Definition von lokalen Laufwerken, Verzeichnissen und Variablen für die aktuelle Stapeldatei.
SHELL	Angabe des DOS-Befehlsprozessors.
SHIFT	Angabe von mehr als 10 austauschbaren Parametern in einer Stapeldatei.

9.2 Systemstart (STARTUP)

Während im DOS Parameter für den Systemstart und Suchpfade in der AUTOEXEC.BAT abgelegt sind, enthält die CONFIG.SYS vom OS/2 eine Reihe dieser Angaben (z.B. Suchpfade über den Befehl PATH). Um bestimmte Programme gleich beim Systemstart auszuführen, bietet OS/2 zwei Möglichkeiten:

1. **STARTUP.CMD**: Eine Stapeldatei zur detaillierten Festlegung des Erscheinungsbilds und zur Steuerung der Reihenfolge der zu startenden Programme.
2. Ordner **Systemstart**: Ein Ordner in dem beim Systemstart auszuführende Programme, in der Regel als Referenz *(Shadow)*, aber auch zu öffnende Ordner und Systemobjekte abgelegt werden.

Reihenfolge der Ausführung.

Bei der Planung der Aktivitäten für den Systemstart ist es hilfreich, die Ausführungsreihenfolge, welche aber auch von der Art der Aktivität abhängt, zu kennen:

1. STARTUP.CMD ausführen.
2. Wiedereröffnen von Ordnern, welche vor dem Systemabschluß (*Shut Down*) geöffnet waren.
3. Ordner Systemstart öffnen. Das wird vom OS/2 automatisch beim Hochfahren gemacht.

Diese Reihenfolge gilt, wie gesagt, nur bei gleichwertigen Aktivitäten, d.h. beim Aufbau einer Vordergrund-PM-Anwendung, gestartet durch die Datei STARTUP.CMD, kann diese durchaus vom Ausführen eines Stapelprogramms des STARTUP-Ordners „überholt" werden.

Ein Befehl PAUSE kann selbstverständlich die Ausführung nachfolgender Befehle verzögern, wenn dieses aus irgend einem Grunde gewünscht ist.

9.2.1 STARTUP-Datei

Die Stapeldatei für die Ausführungen von Anwendungen beim Systemstart heißt **STARTUP.CMD** und beruht im wesentlichen auf dem Stapeldateibefehl **START** . Dieser bietet einige Parameter um die Erscheinungsform der Anwendung beeinflussen zu können:

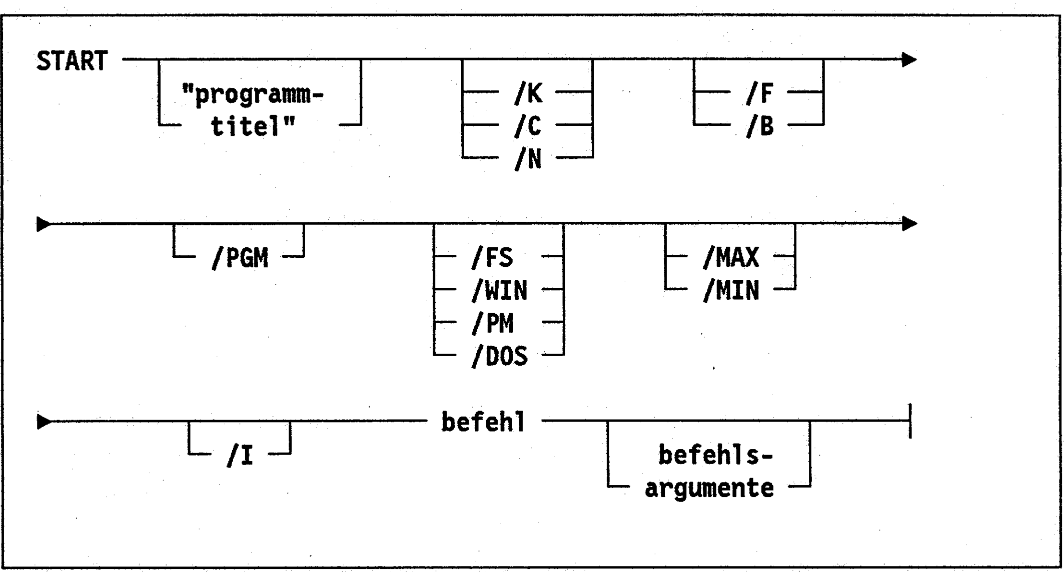


Abbildung 9-1. Syntaxdiagramm Befehl START (J8HTSTRT)

Diese Parameter legen fest, mit welchem Titel die Anwendung in der Aktivitätenliste geführt werden soll, in welcher Erscheinungsform (Vorder-, Hintergrund, vergrößert oder als Symbol), mit welchen Parametern sie ausgeführt wird usw.

Beispieldatei STARTUP.CMD

Die folgende Datei zeigt ein Beispiel für den Aufruf verschiedener Anwendungen bei Systemstart.

Zu bemerken ist, daß der Befehl START eine neue Sitzung öffnet, in welcher das genannte Programm ausgeführt wird. Somit konkurrieren alle mit START geöffneten Programme um Prozessorzeit.

Sollen Programme auf den jeweiligen Ergebnissen eines anderen Programmes aufbauen, so wird der Befehl START nicht verwendet und ein Programm blockiert die Ausführung des nächsten bis zum eigenen Ende.

```
start /max pmddiary.exe
pause
START epm.exe f:\os2work\arbdat00.txt
start "Arbeitsverzeichnis 1:" /f /k /max dir f:\os2work
start "Hintergr.:Textverzeichnis:"  /fs /b dir
f:textwork
```

Abbildung 9-2. Stapeldatei, Beispiel 1

Es werden folgende Aktionen durchgeführt:

- Öffnen des Tagesplaners PMDDIARY in maximaler Größe, weitere Ausführungen erst nach dem Betätigen einer Taste.
- Aufruf des Editors EPM mit einer Arbeitsdatei.
- Anzeigen des Verzeichnisses F:\OS2WORK im Vordergrund in Maximalgröße. kein sofortiges Schließen nach Ausführung.
- Nach Betätigen einer Taste wird das Verzeichnis für F:\TEXTWORK im Hintergrund als Vollbildschirm angezeigt.

Wichtig!

Bei Benutzung von START ist nur die Aufrufs-Reihenfolge festgelegt nicht jedoch die Reihenfolge der Ausführung.

Aus einer CMD-Datei heraus lassen sich keine Systemobjekte anzeigen, weil diese sogenannte Abstrakte Objekte der Arbeitsoberfläche sind und keinen realen Namen haben. Dazu bedarf es eines Programmes, welches das API der Arbeitsoberfläche verwendet.

Dieses Beispiel zeigt, daß die Reihenfolge der Ausführung über den Befehl PAUSE gesteuert werden kann. Geöffnet werden können ausführbare Programme (z.B. PMDDIARY.EXE), nicht aber Ordner oder System-Ordner.

9.2.2 STARTUP-Ordner

Im OS/2 System-Ordner befindet sich der Ordner **Systemstart**; darin abgelegt werden können Ordner, Systemsymbole und ausführbare Programme in Symbolform. Dies geschieht in der Regel als Referenz, damit eine Änderung an der Anwendung, auch beim Systemstart berücksichtigt wird. Soll die Erscheinungsform für ein ausführbares Programm geändert werden oder ein OS/2-Befehl (z.B. DIR) ausgeführt werden, so ist dies über eine Stapeldatei mit dem START-Befehl möglich.

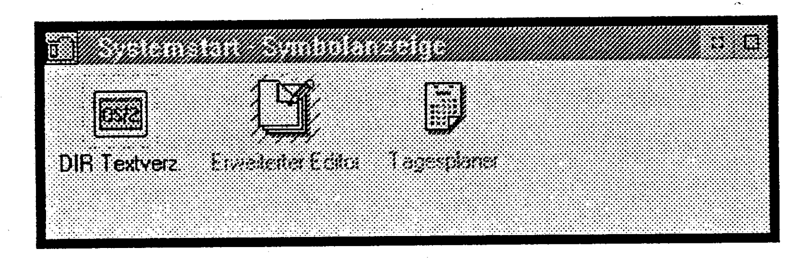


Abbildung 9-3. Der Ordner Systemstart (J8HG0810)

Der Ordner in Abb. 9-3 zeigt, daß eine Anordnung zum Starten von Programmen vielerlei Objekte enthalten kann. Das Objekt DIR ist ein Programmobjekt, welches eine Befehlsdatei aufruft. Um ein solches Objekt zu erstellen, ist folgendes erforderlich:

1. Anlegen eines Programmobjektes von einer Schablone aus und Zuordnen einer schon vorhandenen Befehlsdatei oder
2. Schreiben einer Befehlsdatei mit dem Editor und Sichern im Ordner Systemstart,
3. Ändern des Namens.

Wichtig!

Beim Aufruf von Gesamtbildschirmen aus einem Objekt heraus werden diese möglicherweise in den Hintergrund gelegt. Dann sind sie nur noch über die Programmliste erreichbar. Daher ist es oft besser, OS/2-Fenster zu verwenden.

Studieren Sie das Verhalten solcher OS/2-Sitzungen sehr genau mit Ihren Benutzern, um eine akzeptable Gestaltung zu finden.

Kapitel 10. Grafikfunktionen des PM

Ziele dieses Kapitels

In diesem Kapitel lernen Sie:

- Wie Einheitentreiber und Schriftarten installiert werden
- Welche grafischen Möglichkeiten im PM enthalten sind. Die Kenntnis dieser Möglichkeiten hilft insbesondere beim Definieren von Forderungen an ein Programm bzw. bei der Formulierung von Ausschreibungen
- Weiterhin lernen Sie die Dienstprogramme für Grafik kennen.

Inhalt dieses Kapitels

10.1 Schriftarten unter OS/2 10-3
10.1.1 Installieren von Schriftarten 10-5
10.2 Rastergrafik und Vektorgrafik 10-6
10.3 Grafische Grundfunktionen 10-8
10.4 Darstellungs-Bereich und Koordinatensysteme 10-10
10.5 Grafik-Segmente und Metafiles 10-14
10.6 Einheiten-Kontext 10-15
10.7 Das Dienstprogramm "Bilddateien anzeigen" (PICVIEW) 10-16
10.8 OS/2 Grafik-Anwendung PM Chart 10-17
10.8.1 ÜBUNG: Arbeiten mit dem Programm PMCHART 10-20
10.8.2 Erweiterte ÜBUNG: Malen und zeichnen mit dem PMCHART 10-23

10.1 Schriftarten unter OS/2

Eine Schriftart (*font*) sind Zeichen und Symbole in einer bestimmten Größe und in einem bestimmten Stil, die verwendet werden, um Texte auf Bildschirmen und Druckern zu erzeugen. Bei der Installation des Systems wurden die IBM Schriftarten automatisch installiert, sofern nichts anderes angegeben wurde.

Innerhalb des OS/2-Systems sind verschiedene Arten von Schriftarten verfügbar:

- VIO-Window-Fonts - dienen zur Anzeige der Informationen im OS/2-Fenster und im DOS-Fenster
- OS/2-Bitmap-Fonts - nur in bestimmten Größen verfügbar (z.B.: Helv 8, 10, 12, 18)
- Adobe*-Schriftarten vom Typ 1
- XGA Antialiased Fonts (geglätteter Text nur für XGA)

Problem früher OS/2-Versionen (bis 1.2) und anderer Betriebsystemumgebungen (Windows* 3.0) war es, daß die Schriftarten für Bildschirm und Drucker an verschiedenen Stellen gespeichert waren. So war das Druckerbild oft sehr vom Bildschirm verschieden.
Ein echtes WYSIWYG (*What You See Is What You Get*) war somit kaum möglich. Deshalb hat IBM den ATM (*Adobe Type Manager*) in das SAA-Konzept als integralen Bestandteil mit einbezogen. Der ATM, der zum Betriebssystem gehört, akzeptiert Schriftarten vom Typ 1 und arbeitet mit OS/2-Anwendungen, die den Präsentations-Manager verwenden.

Prinzip des ATM

Bei der ATM Technologie handelt es sich um einheitenunabhängige *„outline fonts"*, welche unabhängig von der verwendeten Hardware je Zeichen als mathematische Formel gespeichert sind. ATM ist ein *„rasterizer"* (Erstellung des Bitmaps aus dem Ergebnis der Formel) für Textausgabe auf dem Bildschirm **und** auf dem Drucker, während POSTSCRIPT gleiches für entsprechende Drucker realisiert. Alle IBM Schriftarten (außer Symbolschriftarten) sind für die Verwendung durch PostScript*-Drucker geeignet.

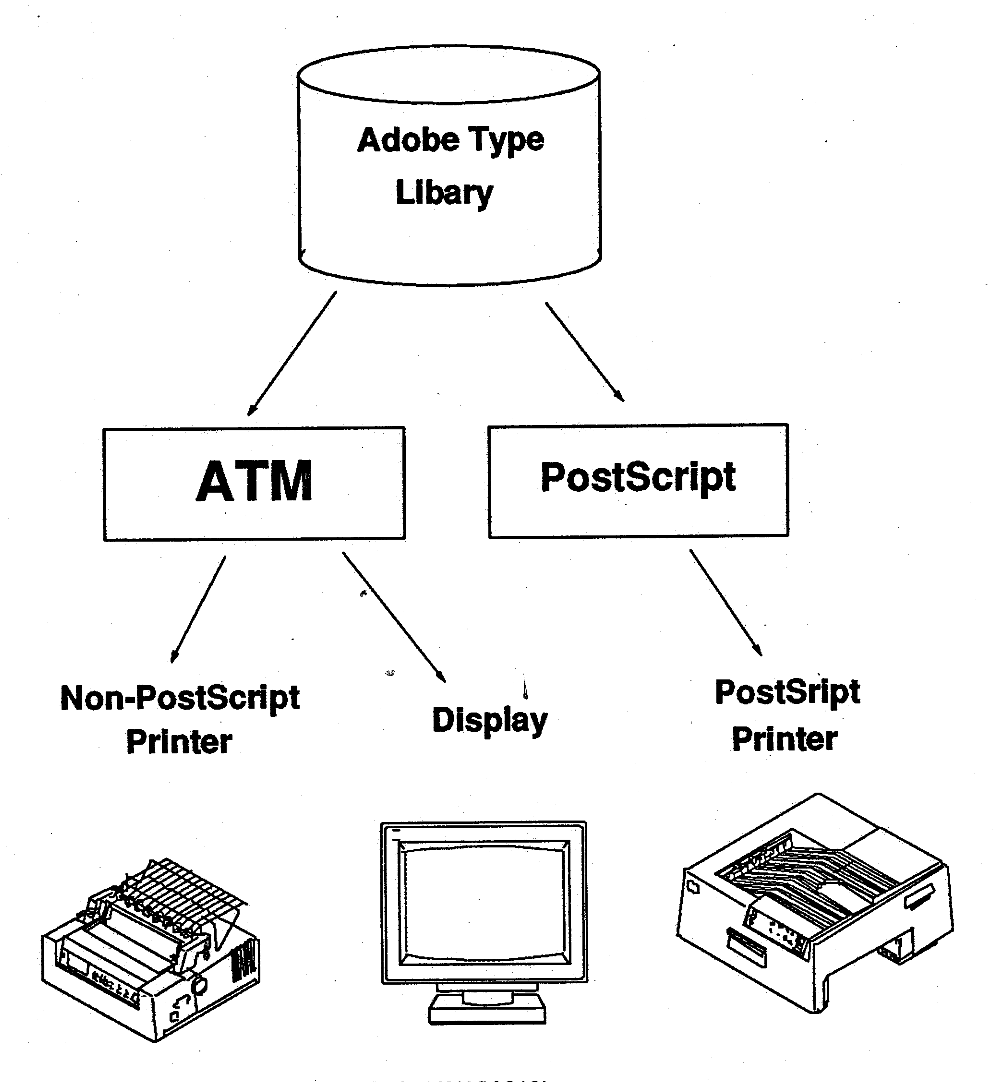


Abbildung 10-1. Prinzip des ATM (J8HG0612)

10.1.1 Installieren von Schriftarten

Mit OS/2 werden folgende Schriftarten geliefert:

- Schriftart Courier und (-Kontur) in mehreren Größen (ATM und Bitmap)
- Schriftart Helvetica/Helv und (-Kontur) in mehreren Größen (ATM und Bitmap)
- Systemschrift mit fester Schrittweite, *System Monospaced*
- Schriftart Times Roman in mehreren Größen (ATM und Bitmap)
- Schriftart Times New Roman und (-Kontur) in mehreren Größen (ATM und Bitmap)

Bitmap-Schriftarten sind als *.FON-Dateien gespeichert. Zusätzliche (Bitmap-)Schriftarten können mit dem Font-Editor und dem *resource compiler* erstellt werden oder am Markt verfügbar sein.

Für die Verwendung von ATM unter OS/2 stehen einige tausend Schriftarten im Format des Typ 1 zur Verfügung. Pro Schriftart sind bei Schriftarten vom Typ 1 immer zwei Dateien mit den Erweiterungen .AFM und .PFB erforderlich. Die ausgewählten Schriftarten werden von der Diskette bzw. aus dem temporären Verzeichnis in das Verzeichnis C:\OS2\DLL kopiert (wenn kein anderes Verzeichnis angegeben wurde).

10.2 Rastergrafik und Vektorgrafik

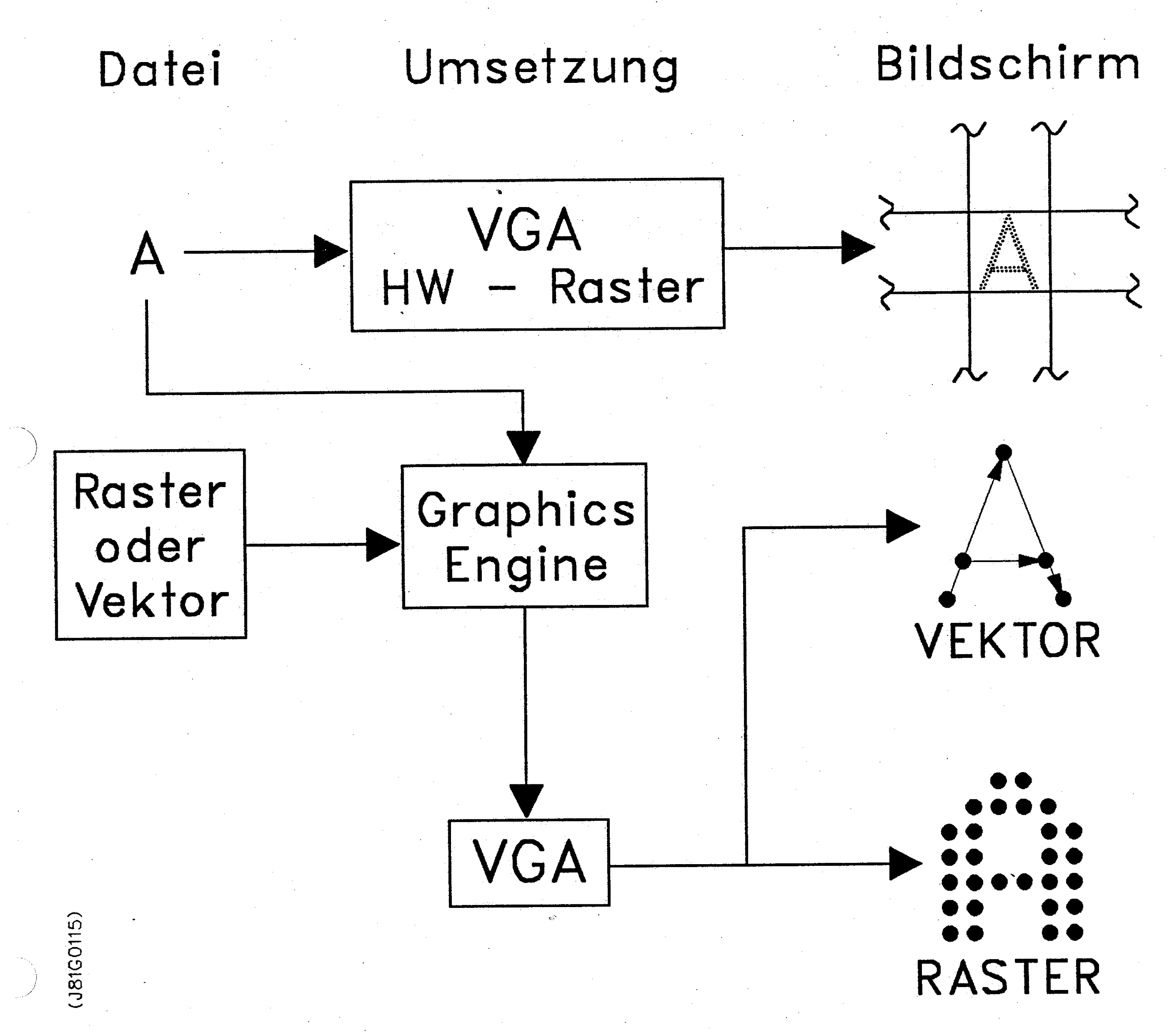


Abbildung 10-2. Der Weg eines Zeichens zum Bildschirm (J8HG1001)

Text nimmt am Bildschirm eine Fläche ein, welche ihm entweder durch Hardware zugeteilt wird und unveränderbar ist oder welche ihm per Programm zugewiesen wird und dann steuerbar ist. Am Bildschirm, der eine Rasteranzeige hat, ist dieser Unterschied nicht so ohne weiteres erkennbar.

Die Zuweisung per Hardware geschieht dadurch, daß ein Text in den Speicher einer Bildschirmsteuerung gestellt wird und diese dann die Zeichen nach einem vorgegebenen Schema zur Anzeige bringt. Meist werden 25 Zeilen je 80 Zeichen angezeigt. Der Vorteil ist, daß die Bildschirmroutine den Text vergessen kann, wenn er einmal in den Anzeigepuffer gestellt wurde, denn dann übernimmt ein Hardware-Zeichengenerator den Rest. Diese Art der Anzeige wird nur bei der Benutzung des Gesamtbildschirmes verwendet.

Ganz anders ist es mit der Grafik. Im Speicher wird ein Anzeigepuffer vorgehalten, der u.U. mehrere Bytes pro Anzeigepunkt (*pel = Picture ELement*) groß ist. Ein Buchstabe wird hier hinein „gemalt". Das kann in jeder beliebigen Schriftart und -größe geschehen und ein Buchstabe kann einige dutzend oder hundert Byte belegen. Entsprechend ist die Software beschäftigt, welche diese Vielzahl von Bytes manipulieren muß.

Bei den sogenannten „soft fonts" werden Rasterfonts und Vektorfonts unterschieden. Rasterfonts sind „vorgefertigt" gespeichert und schnell abrufbar und sie lassen sich leicht anzeigen. Vektorfonts dagegen werden Zeichen für Zeichen „gemalt" und sind ideal für Vergrößerungen, Verkleinerungen, das Füllen mit Farben, Mustern usw. Sie lassen sich im Gegensatz zu den Rasterfonts auch leicht in jedem Winkel und auf jeder Art von Grundlinie zeichnen.

10.3 Grafische Grundfunktionen

Viele Grafiken setzen sich aus Grundelementen, den ***Graphic Primitives*** zusammen. Der reiche Schatz an solchen Funktionen erleichtert das Programmieren von Anwendungen. Das ist für den Anwender bestenfalls an der Geschwindigkeit und der Betriebssicherheit erkennbar.

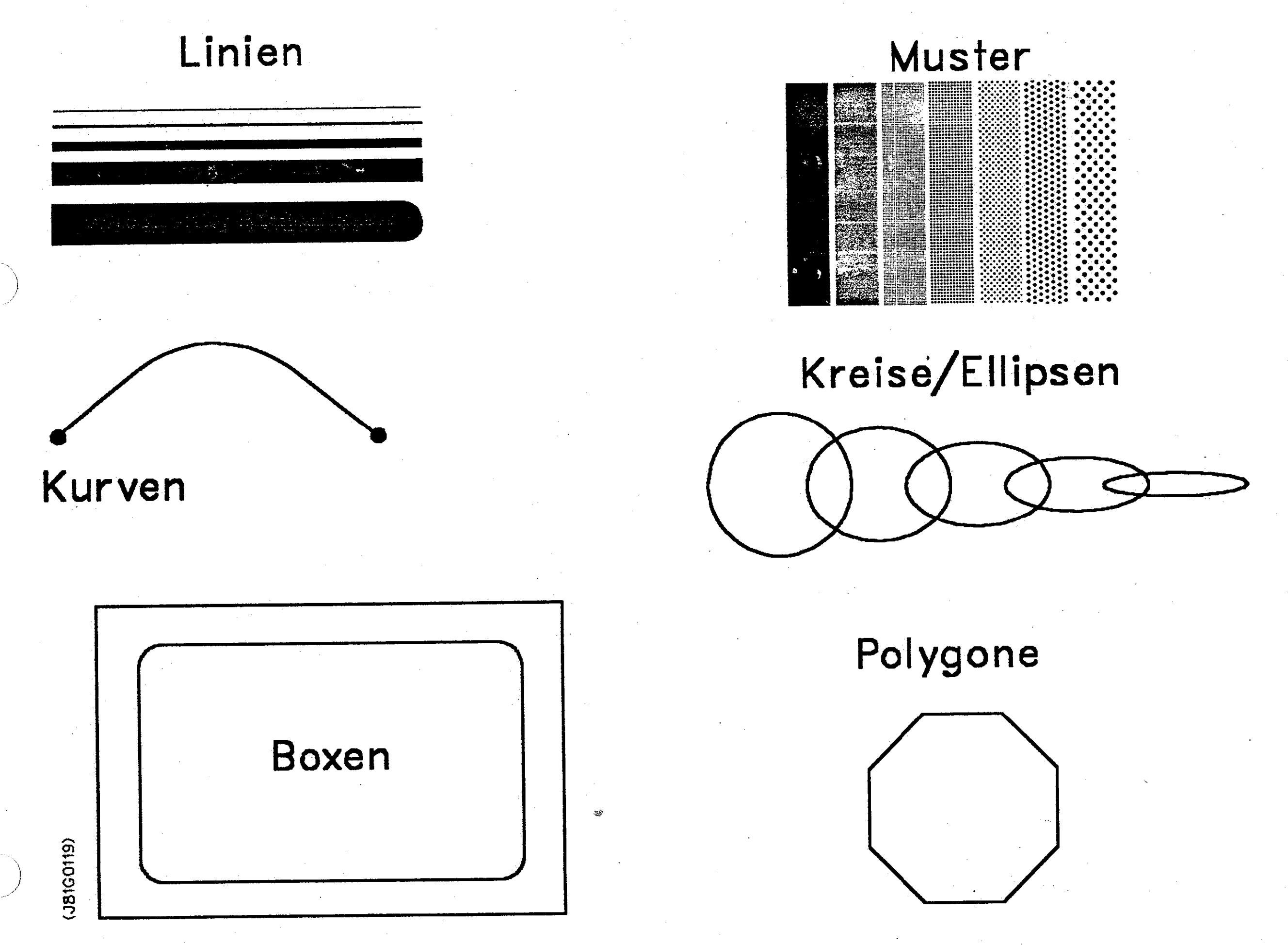


Abbildung 10-3. Grafische Grundfunktionen (J8HG1002)

Neben den hier gezeigten *primitives* gibt es noch Markierungssymbole für Grafiken wie Kreise, Kreuze, Dreiecke usw.. Weiterhin gibt es Linien, welche aus Strichen und Punkten zusammengesetzt sind. Z.B. Strich-Punkt-Strich usw.

Die folgenden Parameter zeigen die von PM gebotenen

Linientypen:

```
LINETYPE_DEFAULT
LINETYPE_DOT
LINETYPE_SHORTDASH
LINETYPE_DASHDOT
LINETYPE_DOUBLEDOT
LINETYPE_LONGDASH
LINETYPE_DASHDOUBLEDOT
LINETYPE_SOLID
LINETYPE_INVISIBLE
LINETYPE_ALTERNATE
```

Linien werden entlang eines **Grafischen Pfades** gezeichnet, welcher nicht sichtbar ist. Dort, wo breite Linien zusammenstoßen, könnten unsaubere Verbindungen entstehen. Das wird durch eine Angabe beim Zeichnen verhindert, welche vorbestimmt, wie die Verbindungen aussehen sollen:

- Gerundete Verbindung
- Eckige Verbindung
- Gefaste Verbindung

PM kann mit den GPI-Funktionen auch häufig Datenstrukturen annehmen, welche dann vom PM in entsprechende Grafik umgesetzt werden.

Die grafischen Funktionen sind für den Programmierer in Form von etwa 200 Funktionsaufrufen verfügbar. Im API des Präsentations-Manager sind viele Funktionen enthalten, die bei anderen Betriebsystemen als „Grafische Erweiterungen" zusätzlich erworben werden können. Wichtig ist die Standardisierung des API, welches dem Umfang von GDDM *(Graphical Data Display Manager)* nahe kommt.

10.4 Darstellungs-Bereich und Koordinatensysteme

Der Präsentations-Manager erlaubt das Definieren von Präsentations-Bereichen mit beliebigen Koordinatensystemen. Ein Koordinatensystem kann einen beliebig großen, gedachten Bereich beschreiben, welcher mit Hilfe des Präsentations-Manager auf die verfügbare Hardware angepaßt wird. Es spielt keine Rolle, ob es sich um Meilen, KM oder Grad und Minuten handelt.

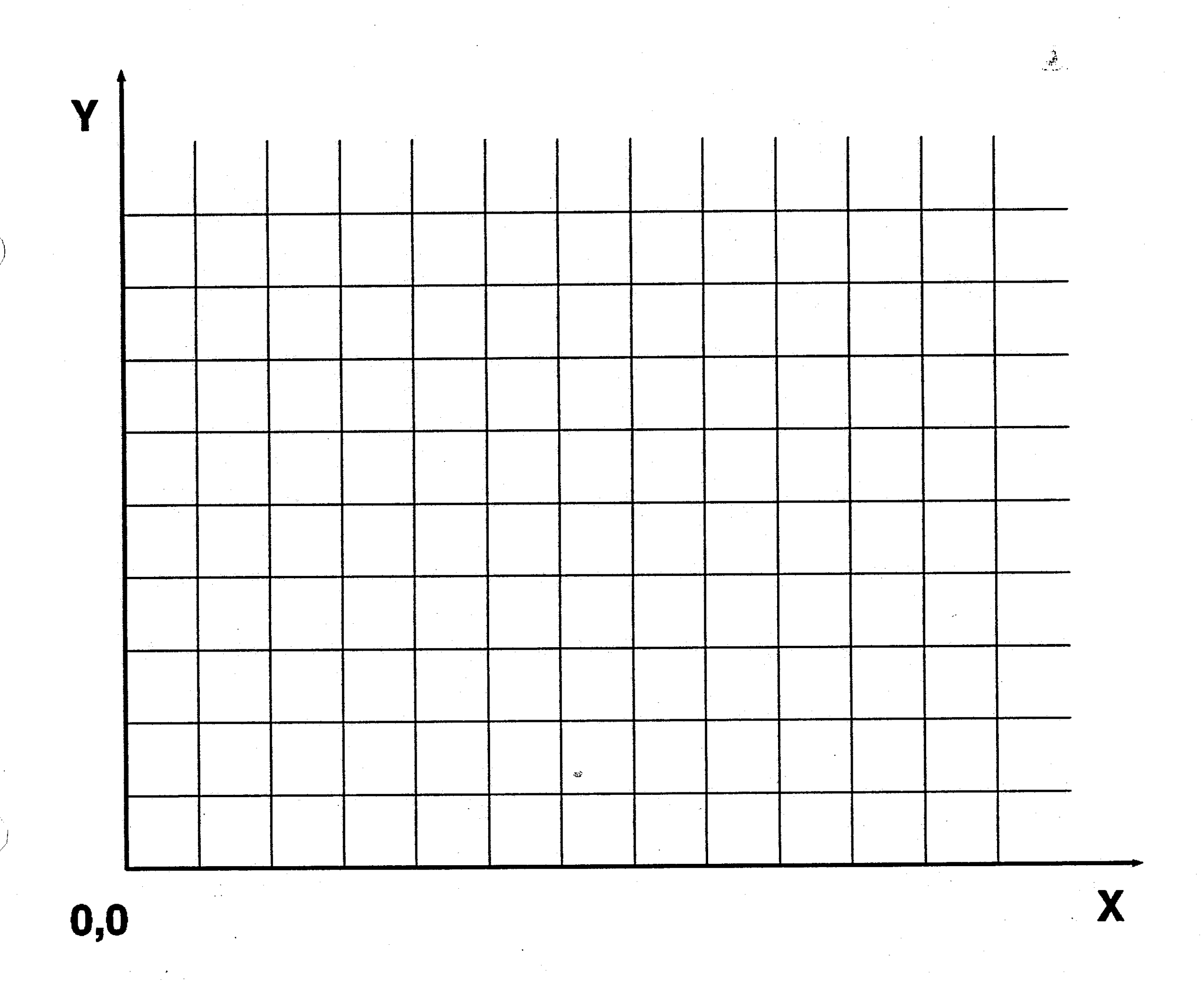


Abbildung 10-4. Hardware-unabhängiges Koordinatensystem (J8HG1003)

Die Koordinaten können im Bereich zwischen -134 217 728... +143 217 727 liegen. Die in der Anwendung gewählten Koordinaten beschreiben den Darstellungsbereich, innerhalb dessen alle Grafik liegen muß.

Im Gegensatz zu grafischen Koordinaten wird bei Texten die für die Datenverarbeitung übliche Anordnung von Zeilen und Spalten vorgegeben. Der Nullpunkt ist oben links am Bildschirm.

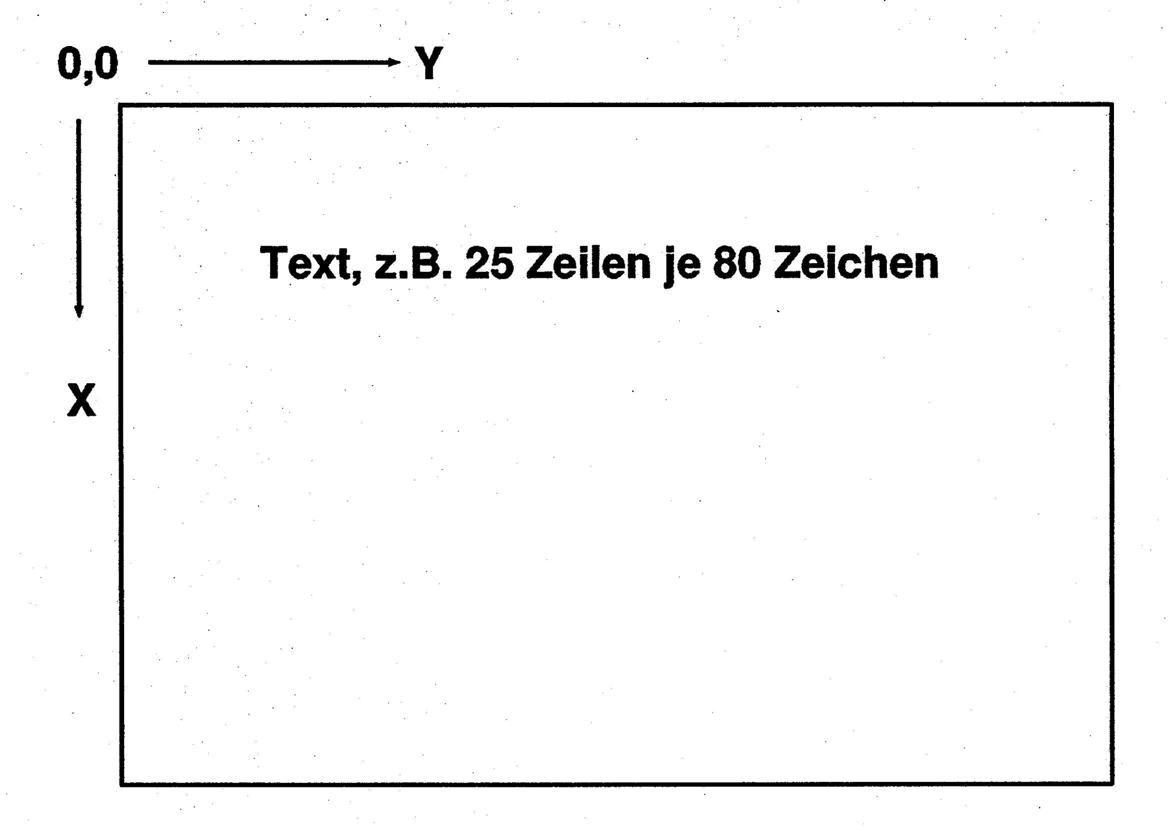


Abbildung 10-5. Hardware-bezogenes Koordinatensystem (J8HG1004)

In bestimmten Fällen kann es sinnvoll sein, eine ganz bestimmte Hardware zu verwenden und die Anzeige darauf abzustimmen. Dann können die Koordinaten des vorhandenen Bildschirmes abgefragt und verwendet werden. Beispiel: Das IBM Modell P75 hat einen XGA-Adapter, zeigt am Bildschirm IBM 8513 mit VGA-Auflösung an und verwendet beim IBM 8515 die volle Leistungsfähigkeit des XGA.

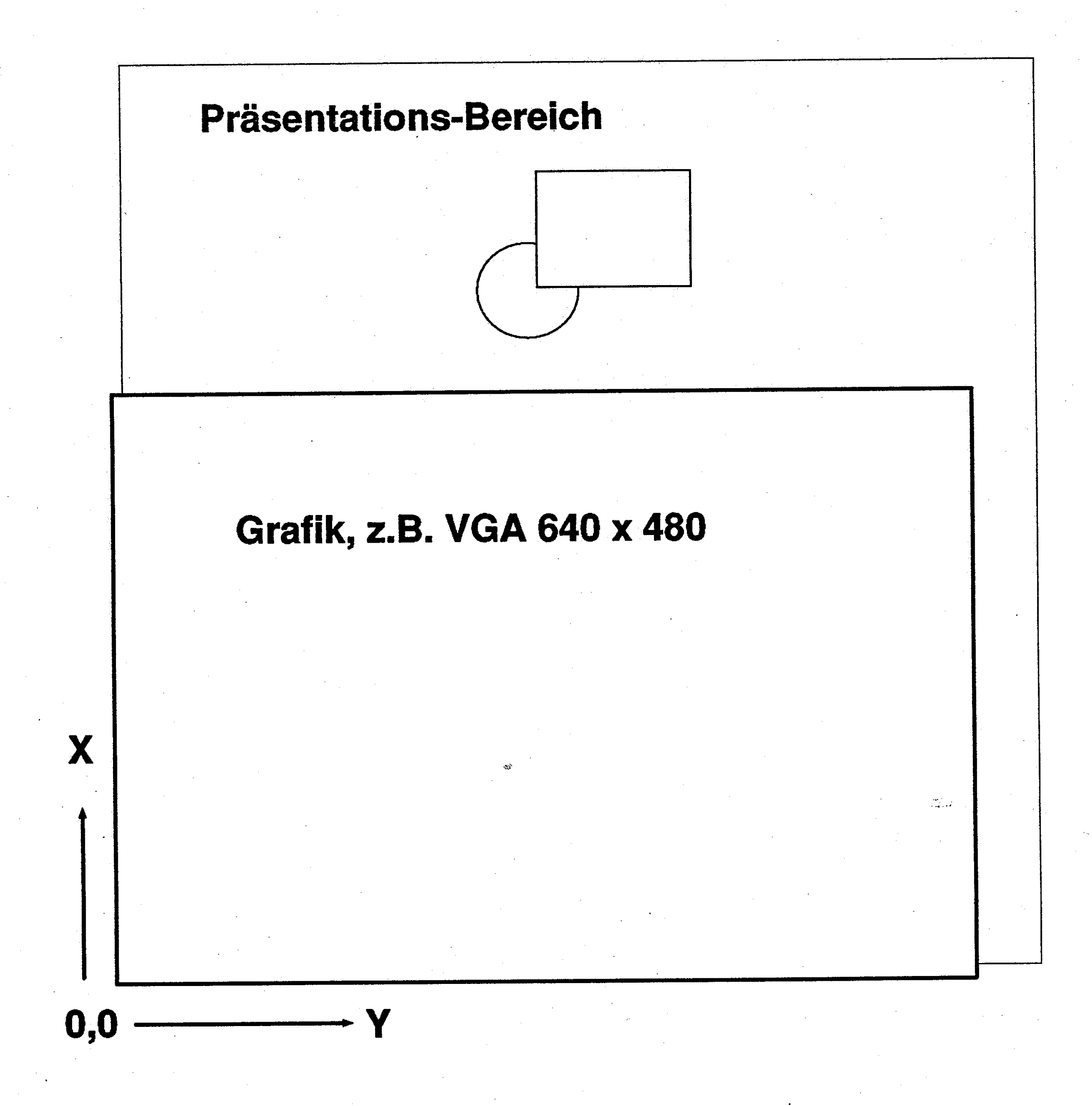


Abbildung 10-6. Hardware-bezogenes Koordinatensystem (J8HG1005)

Ein *presentation space*, welcher größer ist als der Bildschirmbereich, kann nicht ganz angezeigt werden. Wenn das Fenster der Anwendung den Bereich bis X=639 und Y=479 anzeigt, so werden Objekte außerhalb dieser Koordinaten nur durch Verschieben des Anzeigefensters (*viewport*) sichtbar.

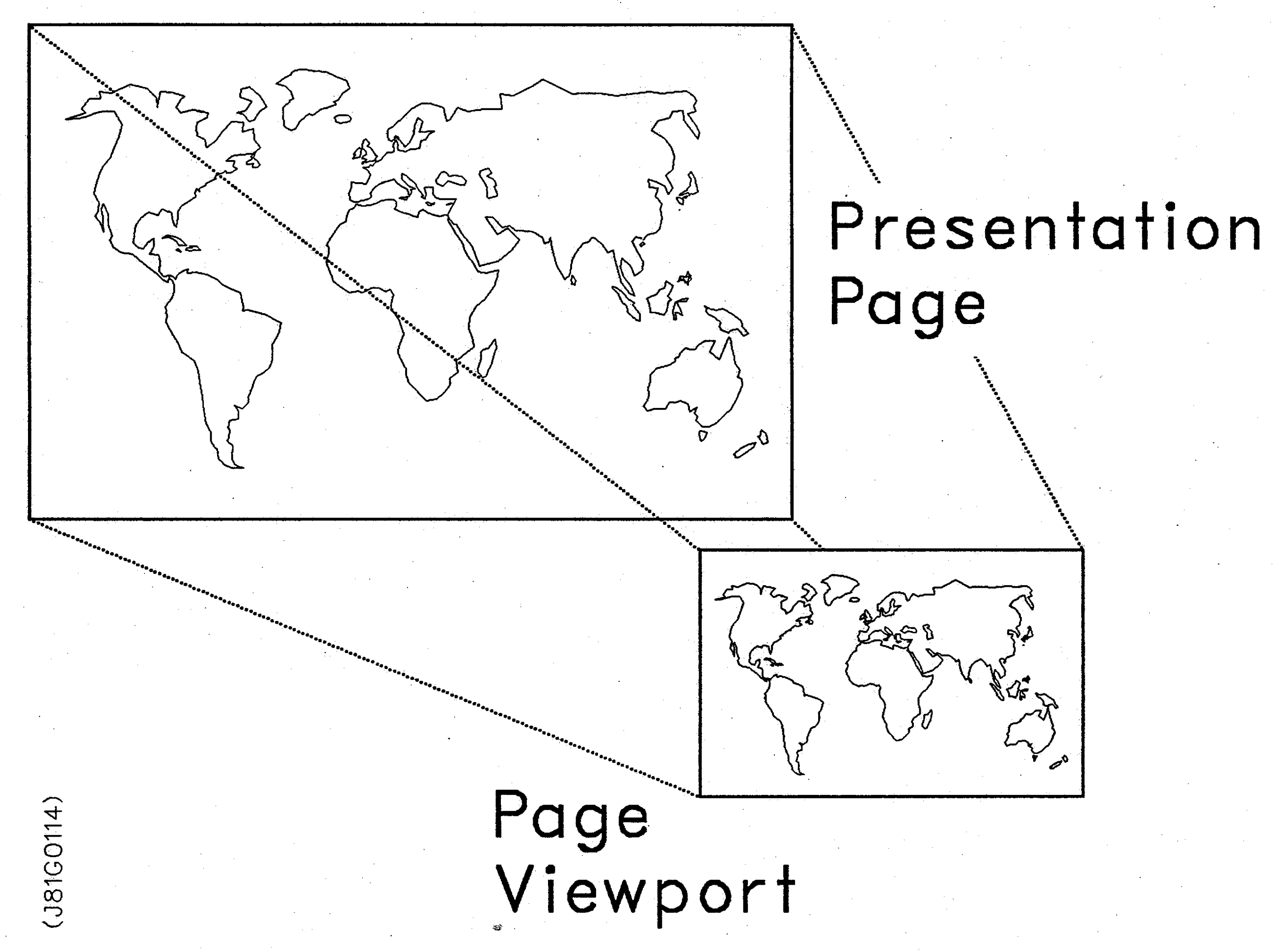


Abbildung 10-7. Presentation Space und Page Viewport (J8HG1006)

Die Weltkarte wurde in einem beliebigen Koordinatensystem beliebiger Größe gezeichnet. Zur Darstellung am Bildschirm muß der *presentation space* **transformiert** werden. Das macht der PM automatisch aufgrund des *device context*.

10.5 Grafik-Segmente und Metafiles

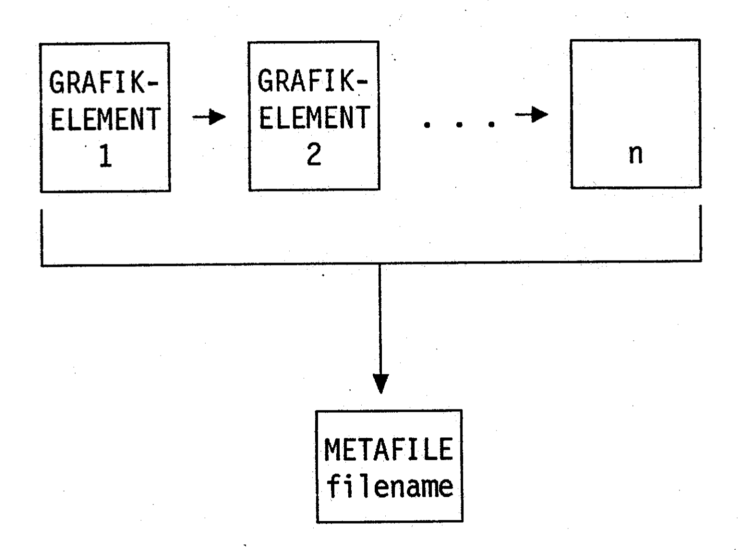


Abbildung 10-8. Grafikelemente (J8HT1001)

Grafische Objekte können als **Grafik-Segmente** zusammengefaßt werden. Diese Segmente bekommen einen Namen, können miteinander verbunden und dann als Datei gespeichert werden. So sind diese Segmente dann Metafiles und sind jederzeit wieder abrufbar und stehen später auch anderen Programmen wieder zur Verfügung.

Das PM-Meta-Format ist nicht mit anderen Formaten vergleichbar, weil in den Dateien die PM-Befehle zum Erstellen einer Grafik enthalten sind und nicht die Grafik selbst. Eine Meta-Datei läßt sich schnell ändern und erneut „abspielen".

10.6 Einheiten-Kontext

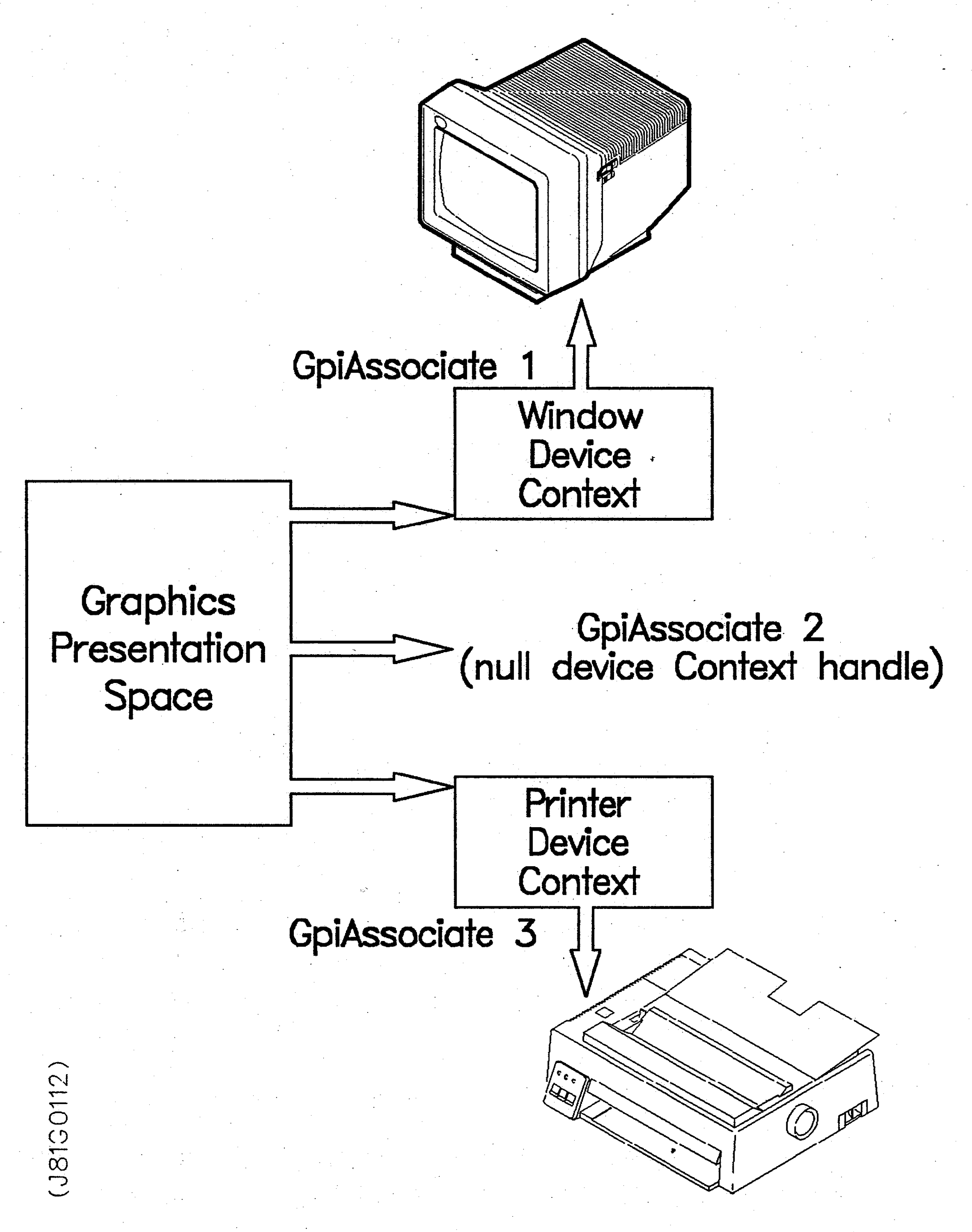


Abbildung 10-9. Zuordnung Anzeigebereich zu Ausgabe-Einheit (J8HG1008)

Grafik wird im Darstellungsbereich aufgebaut und bei Bedarf einer Einheit zugeordnet.
Der Bildschirm ist standardmäßig dem Darstellungsbereich der Anwendung mit Eingabe-„Erlaubnis“ zugeordnet.

10.7 Das Dienstprogramm "Bilddateien anzeigen" (PICVIEW)

Metafiles und .PIF-Dateien (*Picture Interchange Format*) können mit dem Dienstprogramm PICVIEW angezeigt, umgewandelt und gedruckt werden. Das Programm wird bei der Installation des Betriebssystemes automatisch als eine der Produktivitätshilfen installiert.

Die Funktionen des Programmes stehen auch als Funktionsaufrufe zur Verwendung in Programmen zur Verfügung, so daß auch hier eine mächtige Unterstützung für die Programmierung genutzt werden kann.

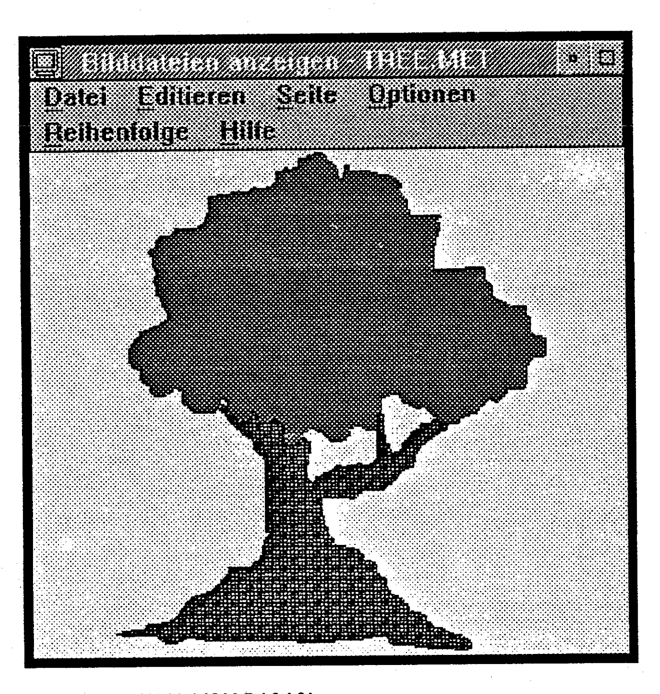


Abbildung 10-10. Dienstprogramm PICVIEW (J8HG1010)

Bilddateien anzeigen zeigt eine vorhandene Grafik in voller Fenstergröße, um eine Beurteilung der Grafik zu ermöglichen. Bei Veränderung der Fenstergröße wird auch die Größe der Grafik angepaßt.

Merke

Das Programm PICVIEW ersetzt die drei Programme PICPRINT, PICSHOW und PICICHG des OS/2 Version 1.3.

10.8 OS/2 Grafik-Anwendung PM Chart

Die grafischen Fähigkeiten des Präsentations-Manager sind in der sehr leistungsfähigen Anwendung PM Chart weitgehend umgesetzt worden. Dieses Kapitel macht Sie mit den wesentlichen Funktionen vertraut. Mit PM Chart können folgende Grafikdateiformate erstellt werden:

Format	Formatname/Herkunft
GRF	Micrografx Charisma Graphic
DRW	Micrografx Draw

Das Format:

Format	Formatname/Herkunft
DAT	Micrografx Charisma Data

kann verarbeitet werden.

Folgende Formate können importiert und dann weiterverarbeitet werden:

Format	Formatname/Herkunft
DIF	Data Interchange Format
WRK	Lotus 123 Version 1.x
SPC	Space Delimited ASCII
SLK	Microsoft Symbolic Link
XLS	Microsoft Excel
WK1	Lotus 123 Version 2.x

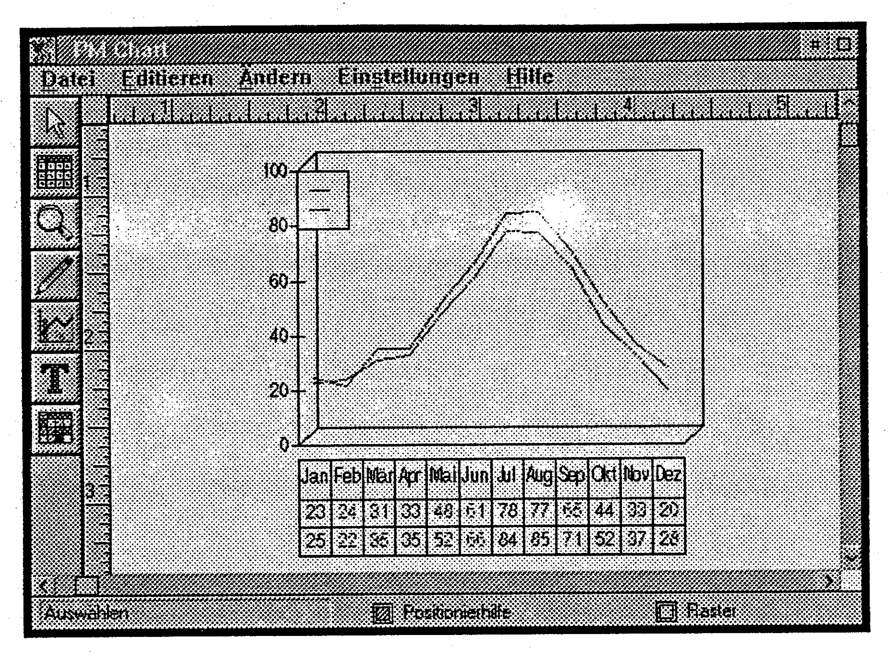


Abbildung 10-11. Dienstprogramm PM Chart mit Liniengrafik (J8HG1020)

Mit PM Chart können Sie folgendes:

1. Grafiken jeder Art erstellen mit:
   - Texten in allen OS/2-Schriftarten und -größen
   - Kreisen und Ellipsen,
   - Rechtecken, auch mit abgerundeten Ecken,
   - Linien

     Freihand

     Gerade

     Bezier

     verschiedene Stärken

     verschiedene Enden
   - Alle Elemente in allen Farben
   - Vielfältige Anordnungsmöglichkeiten
     - Ebenen (Vorder- und Hintergrund)
     - Ausrichten
     - Spiegeln
     - Glätten
     - Drehen:Zusammenfassen
     - Duplizieren

2. Geschäftsgrafiken, einfach und 3D
   - Balkendiagramme
   - Liniendiagramme
   - Tortendiagramme
   - Flächendiagramme
3. Erstellen von Tabellen
4. Importieren von Tabellen von vielen Programmen

10.8.1 ÜBUNG: Arbeiten mit dem Programm PMCHART

Ziel der Übung
Sie erfahren, wie Sie mit diesem Programm schnell eine Zahlengrafik erstellen können.

Durchführung

1. Den PM Chart finden Sie im Ordner Produktivität.
2. Aktivieren Sie das Objekt PMCHART.
3. Beginnen Sie mit der Auswahl Datei, Neu
4. Bevor Sie eine Grafik erstellen können, benötigen Sie erst einmal Ihre Werte:
   - In der linken Spalte des PM Chart finden Sie 7 verschiedene Symbole.
   - Wählen Sie das zweite, nämlich das Symbol für Tabelle aus.
   - In das sich nun öffnende Arbeitsblatt geben Sie nun in die Spalte „A" ein:

     ```
     Stühle
     Tische
     Bänke
     Regale
     ```
   - In die Spalte „B" geben Sie z.B. ein:

     ```
     345
     421
     123
     230
     ```
   - In die Spalte „C" geben Sie z.B. ein:

     ```
     567
     543
     876
     435
     ```
   - Markieren Sie bitte den ganzen Bereich Ihrer Eingaben.
   - Blenden Sie durch Anklicken des Tabellen-Symbols die erstellte Tabelle aus.
5. Jetzt wählen Sie bitte das Symbol für „Grafik" aus. Es ist das fünfte von oben.

- Klicken Sie bitte gleich die erste Variante, die senkrechte Blockgrafik an.
- In dem sich öffnenden Dialogfenster wählen Sie bitte aus: 3D, Legende und Tabelle bestätigen mit Neu.

6. Über Ändern, Farben/Stil, Farbe und Diagrammfarben können Sie durch Auswählen einer Farbe ein neues Farblayout für die nächste Grafik erstellen.
7. Wenn Sie darüber hinaus das Feld Festlegen aktivieren, können Sie auch die Farben jeder einzelnen bereits erstellten und markierten Grafik verändern.
8. Stellen Sie mit Editieren und Löschen sicher, daß Sie nur eine Grafik auf dem Bildschirm haben.
9. Nun beschriften Sie Ihr Blatt noch:
   - Wählen Sie das Symbol „T“ aus.
   - Mit ABC können Sie eine Schriftart auswählen.
   - Mit abc definieren Sie dann den Modus für das Beschriften.
   - Nun können Sie loslegen. Klicken Sie die Stelle auf dem Bildschirm an, an der Ihr Text beginnen soll und geben dann den gewünschten Text ein:

     `Möbelhaus MBI`
10. Jetzt klicken Sie das erste der Symbole an, den Pfeil. Damit stellen Sie wieder auf den normalen Bearbeitungsmodus um.
11. Klicken Sie Ihren Schriftzug an und ziehen Sie ihn mit der Maus in die gewünschte Größe.
12. Markieren Sie den Schriftzug, wählen Sie Ändern und Drehen und versuchen Sie, ob eine etwas verdrehte Darstellung des Schriftzuges Ihre Grafik möglicherweise besser zur Geltung bringt.
13. Jetzt geben Sie einen neuen Text ein:

    `Verkaufszahlen der Jahre 1991 und 1992`

    und stellen Sie ihn dort hin, wo sie ihn haben möchten.
14. Unter Umständen möchten Sie auch Ihrer Tabelle die Jahreszahlen zufügen.
15. Wenn Sie jetzt einen ganzen Bereich, z.B. die Grafik einschließlich Tabelle und Beschriftung noch auf Ihrem Blatt verschieben möchten, wählen Sie Editieren und Blockauswahl, setzen den Mauszeiger an den Anfangspunkt des zu markie-

renden Blocks und ziehen bei gedrückter Taste soweit, daß der benötigte Bereich alle gewünschten Elemente einschließt.

16. Nun klicken Sie den Block einmal an und können jetzt nach Belieben verschieben.

17. Für den Fall, daß Sie Ihrer Grafik eine Farbe unterlegen wollen, ist es nicht zu spät dazu.

    - Wählen Sie Einstellungen
    - Anzeigenfarbe
    - Klicken Ihre Farbe an und bestätigen mit OK.
        — Über Palette haben Sie die Möglichkeit aus einem größeren Farbangebot auszuwählen.

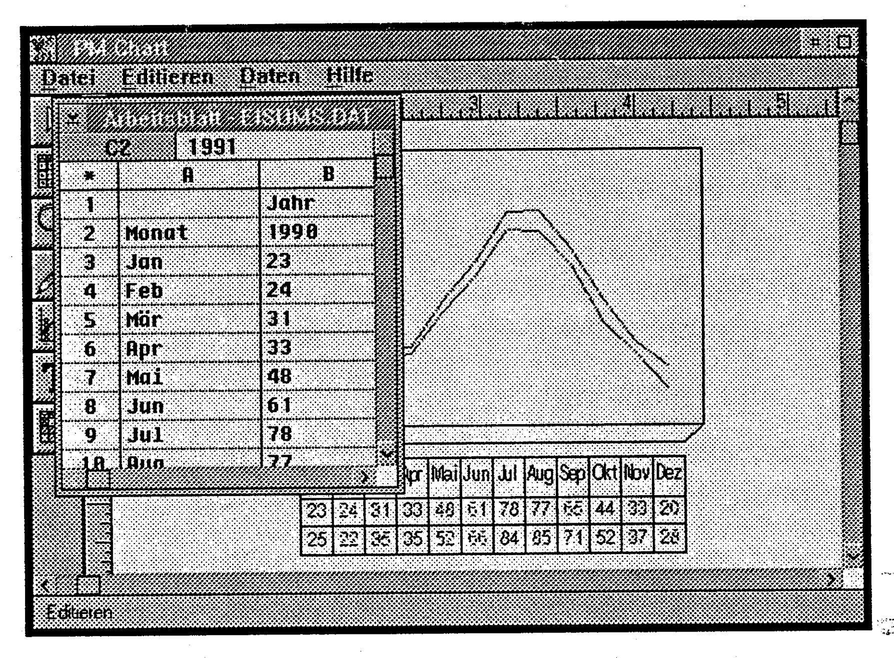


Abbildung 10-12. PM Chart mit Tabelle und Grafik (J8HG1022)

10.8.2 Erweiterte ÜBUNG: Malen und zeichnen mit dem PMCHART

Ziel der Übung

Sie erfahren, wie Sie mit diesem Programm malen und zeichnen können.

Durchführung

1. Beginnen Sie wieder mit Datei Neu.
2. Als erstes gestalten Sie eine Fläche in abgestuften Farben.
   - Klicken Sie das Bleistiftsymbol an und wählen Sie das Symbol für das Rechteck.
   - Dann setzen Sie den Mauszeiger in die linke obere Ecke der Arbeitsfläche und ziehen Sie das Rechteck weit nach rechts unten.
   - Markieren Sie jetzt die Rechteckfläche mit einem Klick der linken Maustaste.
   - Wählen Sie jetzt das Symbol für die Farbpalette, den Menüpunkt Farbe und dann Verlauf.
   - Definieren Sie jetzt Anfang, wählen Sie Ihre Farbe und bestätigen Sie mit OK. Jetzt definieren Sie Ende, wählen Sie erneut eine Farbe. und bestätigen Sie wieder mit OK.
   - Bislang hat sich noch nichts sichtbares getan, wählen Sie daher den Menüpunkt Stil und wählen Sie das Muster, das Ihnen am meisten zusagt. Probieren Sie ruhig alle 4 Muster aus.
3. Probieren Sie jetzt einige Möglichkeiten aus mit Formen und Strichstärken zu arbeiten:
   - Klicken Sie wieder Farbe und Farbenstil an.
   - Dann aktivieren Sie den Radioknopf Linie und danach den Menüpunkt Stil.
   - Sie haben die Möglichkeit, verschiedene Linienarten, Strichstärken und Linienenden zu definieren.
   - Suchen Sie sich „Ihre“ Elemente aus.

4. Jetzt wählen Sie das Zeichensymbol für die Ellipse aus und ziehen Sie mit der Maus eine Ellipse auf Ihre Fläche.

5. Als nächstes nehmen Sie das Element „Gerader Strich". Gehen Sie wieder in die Farbpalette und definieren Sie z.B. das Strichmuster mit den beiden Pfeilen und ziehen Sie einen Strich auf der Arbeitsfläche, Sie können ihn anschließend vergrößern, verkleinern, verschieben oder schrägstellen.

6. Wählen Sie jetzt die „Kurvenlinie" und ziehen Sie eine Gerade, lassen Sie die Maustaste los, bewegen Sie den Mauszeiger an eine andere Stelle neben der gezogenen Linie und klicken Sie einmal, gehen Sie nun zum Anfangspunkt Ihrer Linie zurück und klicken Sie erneut.

7. Das nächste Symbol steht für eine Freihandlinie. Wählen Sie eine neue Farbe und ziehen Sie eine solche Linie.

8. Der Bogen ist das letzte Element, das wir hier noch ausprobieren möchten. Wählen Sie erneut eine andere Farbe und Strichart, klicken Sie an einer freien Stelle an, halten die Maustaste gedrückt und ziehen den Mauszeiger so, daß sich ein Bogen bildet. Nun lassen Sie die Maustasten los.

Über den Menüpunkt Palette können Sie noch eine Vielzahl weiterer Farben definieren. Sie haben jetzt einige wesentliche Elemente des PM Chart kennengelernt, ich wünsche Ihnen viel Spaß und Erfolg damit bei Ihrer täglichen Arbeit. Mithilfe des Symbols der „Lupe" können Sie sich z.B. auch Einzelheiten herausgreifen, vergrößern.

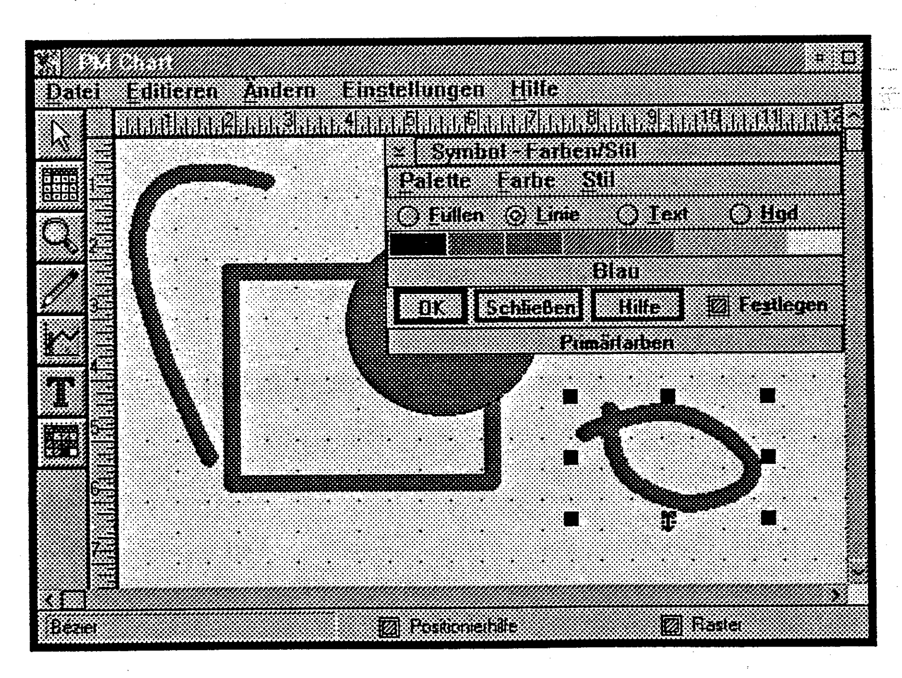


Abbildung 10-13. PM Chart mit Zeichnung (J8HG1024)

Kapitel 11. Überblick Anwendungsentwicklung

Ziele dieses Kapitels

In diesem Kapitel lernen Sie kennen, wie Programme für OS/2 Version 2 erstellt werden, welche Voraussetzungen erfüllt sein müssen und worauf besonders zu achten ist.

Referenzen

- ITSC Red Book IBM OS/2 Version 2.0, Volume 3: Presentation Manager and the Workplace Shell, GG24-3732
- ITSC Red Book IBM OS/2 Version 2, Volume 4: Application Development, GG24-3774
- ITSC Red Book A Practical Introduction to OO Programming, GG24-3641-01

Inhalt dieses Kapitels

11.1 Erweiterter Editor EPM ... 11-3
11.1.1 Übung: Arbeiten mit dem erweiterten Editor "EPM" ... 11-6
11.2 Voraussetzungen Software ... 11-8
11.3 TOOLKIT - Programming Tools and Information ... 11-9
11.3.1 Der Font Editor ... 11-10
11.3.2 Der Dialog Editor ... 11-12
11.4 Informations-Entwicklung ... 11-13
11.4.1 Konzept der Informations-Darstellung ... 11-15
11.4.2 Schreiben einer IPF-Datei ... 11-16
11.4.3 Übung: Verwendung von IPF-Tags ... 11-17
11.4.4 Übung: Hilfedatei mit geteilten Fenstern ... 11-19
11.4.5 Umwandeln einer Informations-Datei ... 11-22
11.4.6 Übung: Erstellen einer Referenzdatei mit IPFC 11-23
11.5 IPF-Tag-Übersicht ... 11-26
11.6 Programmierung für OS/2 ... 11-29
11.6.1 Programmierung zeichenorientierter Schnittstellen ... 11-29
11.6.2 Struktur von PM-Programmen ... 11-29
11.6.3 Programmieren mit SOM ... 11-32

11.7 Unterstützende Programme 11-35
11.8 Die Prozedurensprache REXX 11-36
11.8.1 Überblick 11-36
11.8.2 Voraussetzungen zur Benutzung von PMREXX 11-37
11.8.3 Struktur und Syntax 11-38
11.8.4 Ausdrücke und Operatoren 11-40
11.8.5 Anweisungen (Instruktionen) 11-41
11.8.6 Standard-Routinen 11-44

11.1 Erweiterter Editor EPM

Der erweiterte Editor ist ein PM-Programm, mit dem in mehreren Fenstern verschiedene Textdateien erstellt und editiert werden können. Neben einer Makrosprachenunterstützung bietet er zahlreiche Einstellungsmöglichkeiten, die von Änderungen der Tastaturbelegung über das Einbetten von Compilerfehlern bis hin zum formatierten Druck reichen.

Start des EPM-Editors mit Optionen

Der Editor ist in den Ordner Produktivität als Programmobjekt eingetragen und kann somit einfach gestartet werden. Es kann aber auch sinnvoll sein, dieses Programm über die Befehlszeile oder über eine Kommandoprozedur zu starten, da eine Vielzahl von Parametern und Argumenten unterstützt werden.

Auswahl möglicher Optionen:

- **Zeichenkette in Anführungszeichen** - Angabe eines Editorbefehls
  Beispiel:

  ```
  EPM mbrief.dok 'bottom'
  ```

- **Dateiname** - Angabe von zu ladenden Dateien
  Beispiele:

  ```
  EPM meindat.txt d:\brief\deindat.txt e:\cpm\hilfe.scr
  EPM *.dok d:\test\temp*.*
  ```

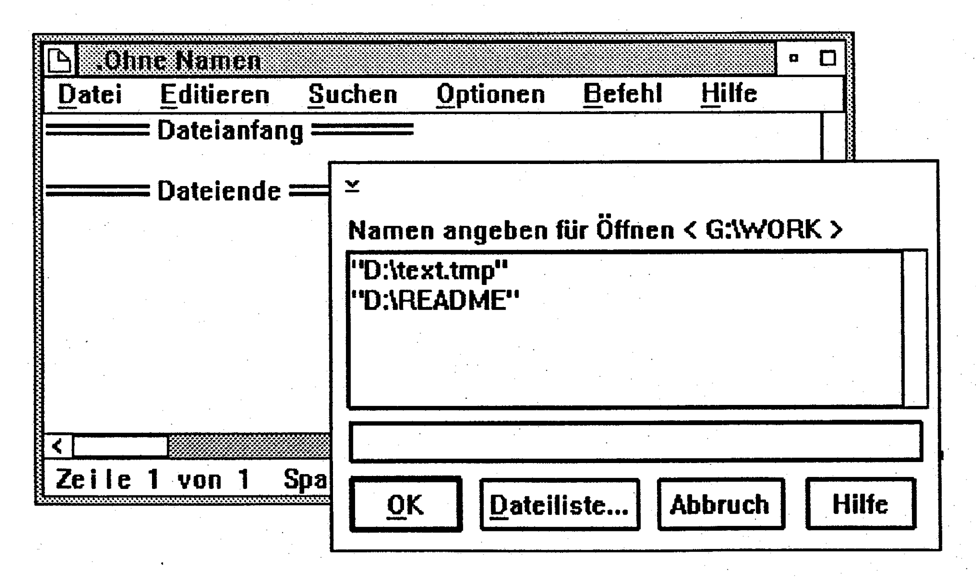


Abbildung 11-1. Erweiterter Editor (J8HG0905)

Hinweise zur Nutzung des EPM

- Nutzen Sie die vielfältigen Möglichkeiten der Hilfeunterstützung
- Wichtige Grundeinstellungen sind im Menü Optionen, Anpassung, Einstellungen... änderbar (Schriftarten, Ränder, Farbe usw.)
- Zwischenspeicher und Blockmanipulationen sind im Menü Editieren verfügbar
- Druck als reiner Text oder als formatierter Text (WYSIWYG) möglich
- Nutzen Sie die Möglichkeiten der Bedienung mit der Maus (Direktmanipulation: Ziehen von einem Dokument in ein anderes)
- Zahlreiche Editorbefehle und andere Einstellungen verfügbar

Auswahl von Tastaturbelegungen

F1	Anzeigen von Hilfetexten
F2	Sichern der aktuellen Datei
F3	Verlassen der aktuellen Datei
F5	Öffnen eines neuen Editierfensters
F6	Zeichnen von Linien in der aktuellen Datei
F7	Umbenennen der aktuellen Datei
F8	Editieren einer Datei
F9	Widerrufen
F11	Umschalten zur vorhergehenden Datei in der Umlaufliste
F12	Umschalten zur nachfolgenden Datei in der Umlaufliste
Alt+A	Versetzen der markierten Textstelle
Alt+B	Markieren der aktuellen Cursorposition
Alt+C	Kopieren des markierten Texts
Alt+D	Löschen des markierten Texts
Alt+J	Zusammenfügen der Zeile mit der folgenden an der aktuellen Cursorposition mit einer Leerstelle dazwischen
Alt+L	Markieren der Zeile, die den Cursor enthält
Alt+M	Verschieben der aktuellen Markierung
Alt+P	Neuausrichten eines Abschnitts nach der aktuellen Randeinstellung (von der Cursorposition an)
Alt+R	Neuausrichten eines markierten Blocks mit Text nach einer neuen Randeinstellung
Alt+T	Zentrieren des Texts in der aktuellen Markierung (oder nach den Randeinstellungen, wenn keine Markierung vorhanden)
Alt+U	Aufheben der aktuellen Markierung

11.1.1 Übung: Arbeiten mit dem erweiterten Editor "EPM"

Ziel der Übung

Sie lernen einige der vielen Möglichkeiten kennen, die Ihnen dieser Editor bietet.

Durchführung

Öffnen Sie eine OS/2 Befehlszeile Geben Sie ein:

```
epm config.sys autoexec.bat help.bat readme
```

Nach dem der Editor gestartet ist, wählen Sie aus: Optionen, dann Anpassung. In dem sich nun öffnenden Dialogfenster wählen Sie nunmehr aus: Umlauffunktion Als Ergebnis dieser Aktion sehen Sie rechts oben neben den „Minimize-" und „Maximize-"Knöpfen sogenannte „Umlaufknöpfe" Probieren Sie bitte deren Funktion aus.

Was ist das Ergebnis?

- ______________________________

Im nächsten Schritt wählen Sie bitte Datei,
Datei hinzufügen... und wählen Sie die Datei HELP.CMD aus dem Verzeichnis \OS2 aus. Jetzt klicken Sie bitte erneut einen der Umlaufknöpfe an und stellen Sie fest, ob nun alle insgesamt ausgesuchten Dateien „hintereinander" liegen.

Nachdem das geklappt hat, markieren Sie bitte einen beliebigen Textteil mit der Maus. (Maustaste 1 vor das erste Zeichen stellen, Maustaste halten und die Maus nach rechts ziehen bis der gewünschte Bereich markiert ist)

Wählen Sie nun Stil..., suchen Sie sich eine andere Schriftart aus und evtl. auch eine andere Schriftgröße aus. Bestätigen Sie Ihre Auswahl mit Festlegen

Was ist das Ergebnis?

- ______________________________

Damit wir uns für die nächste Aktion nichts zerstören kopieren Sie bitte die AUTOEXEC.BAT mit dem neuen Namen BUTOEXEC.BAT. (Entweder aus einer Befehlszeile heraus oder auch aus dem Objekt Laufwerke heraus). Fügen Sie diese nun auch dem EPM hinzu. Geben Sie nun einige Änderungen mehrzeilig ein, wählen Sie Editieren, Widerrufen aus und bewegen Sie im sich öffnenden Dialogfenster den Schieber nach links und rechts. Beobachten Sie dabei Ihre Texteingaben. Was stellen Sie fest?

• ____

Nun wählen Sie Suchen aus, suchen Sie z.B. nach dem Begriff „DOS“ und setzen Sie als Ersatzbegriff „das“ Markieren Sie Alle Textstellen ändern und klicken Sie die Taste Ändern und Suchen an.

Wenn alles geklappt hat, müssen Sie in Ihrer „BUTOEXEC.BAT“ überall da wo der Begriff DOS gestanden hatte, jetzt den Begriff „das“ stehen haben. In einer funktionierenden Datei wäre das sicher nicht sinnvoll, aber diese Datei haben wir ja zum Spielen erstellt.

Für den Fall, daß Sie mal wissen möchten, welche Dateien Sie eigentlich alle in den „Umlauf“ gestellt haben, wählen Sie Optionen, Umlaufliste.

Nun noch einen Tip, wenn Sie sich mit dem „EPM“ noch weiter vertraut machen wollen, vergessen Sie nicht die installierte EPM-Hilfe zu verwenden.

Viel Erfolg mit Ihrem EPM

11.2 Voraussetzungen Software

Die Programmierung erfolgt in einer der folgenden Programmiersprachen:

- IBM C Set/2
- IBM REXX/2
- IBM Assembler/2
- IBM COBOL/2 (ab Version 1.2)
- IBM FORTRAN/2 (ab Version 1.2)
- IBM EASEL
- IBM ENFIN/2
- Smalltalk
- C++

Seit der Version 2 wird eine komplette Arbeitsumgebung zur Programmentwicklung angeboten. Diese umfaßt:

1. IBM Workframe/2, ein Programm zur Steuerung und Organisation der Programmentwicklung
2. IBM C Set/2 32-Bit-Compiler

   er beinhaltet den:

   - PM-Debugger
3. Toolkit

Unter dem Workframe/2 ist jeder beliebige Editor verwendbar.

Der Bedarf an Plattenplatz ist bei mehreren Produkten davon abhängig, welche Teile installiert werden. Ein Richtwert für Workframe/2, einen Compiler wie IBM CSet/2 und für das Toolkit ist je nach Umfang der Installation bis zu etwa 35 MB.

11.3 TOOLKIT - Programming Tools and Information

Das Schreiben von PM-Programmen erfordert nicht nur das Vorhandensein eines geeigneten Compilers sondern auch das *TOOLKIT* entsprechend der Version des Betriebssystems. Seit der Version OS/2 1.2 wurde das *TOOLKIT* beträchtlich erweitert und beeinhaltet jetzt auch das *Technical Reference*. Wegen des erweiterten Umfanges heißt das Produkt jetzt: ***Programming Tools and Information.*** Aus Gewohnheit und wegen der Kürze des Begriffes wird jedoch nach wie vor vom *TOOLKIT* gesprochen.

Wichtig!

Das *Toolkit* muß immer zu der jeweiligen Version des OS/2 passen, weil sonst möglicherweise Funktionen des OS/2 Version 2 angesprochen werden, welche aber vom *Toolkit* nicht unterstützt werden - oder umgekehrt.

Das TOOLKIT beinhaltet eine Reihe von Komponenten, welche Voraussetzung für die Programmierung sind:

- Dialog-Editor und Font-Editor
- Bibliotheken für die Funktionsaufrufe
- Beispielprogramme in C, Assembler, FORTRAN , COBOL und C in Kombination mit REXX
- Resource Compiler
- SOM Compiler
- IPF-Compiler

11.3.1 Der Font Editor

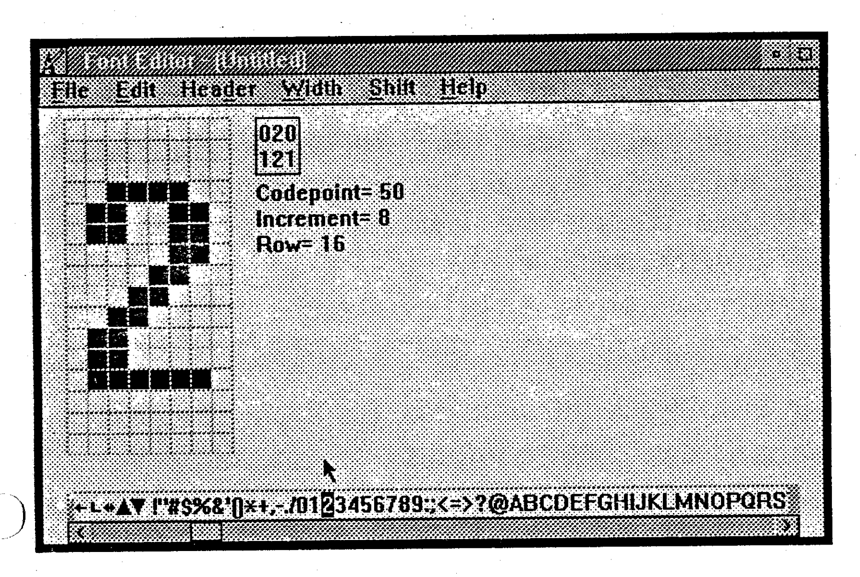


Abbildung 11-2. Font Editor (J8HG1101)

Der Font Editor dient dem Erstellen von Zeichen in einer Fülle von Schriftarten. Die Schriften lassen sich mit fester oder proportionaler Schrittweite erstellen. Eine Vielzahl von Eigenschaften stehen für fast unbegrenzte Gestaltungsmöglichkeiten zur Verfügung.

Neben diesem Hauptzweck lassen sich die Zeichen auch in die Zwischenablage geben, von wo aus sie für Programme wie den Symboleditor erreichbar sind.

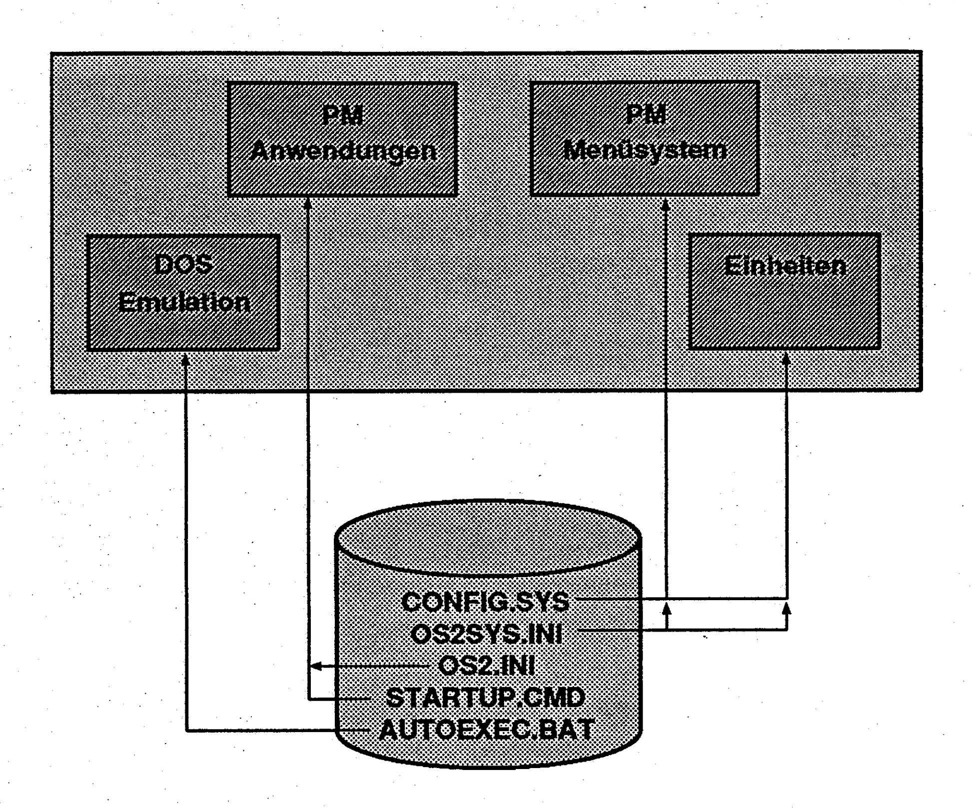


Abbildung 11-3. Konzept Erstellung Schriftarten-Datei (J8HG1102)

Der *Font Editor* kann Dateien mit der Erweiterung .FNT einlesen. Ist keine solche Datei vorhanden, so kann sie mit dem Editor erstellt werden. Der Editor hat einen kompletten ASCII-Zeichensatz, der nach Belieben in jeder Hinsicht verändert werden kann.

Über eine Steuerdatei .RC wird ein Zeichensatz dem *Resource Compiler* zugeführt und dann in eine .FON-Datei umgewandelt.

Statt der WYSIWYG-Methode kann eine Schriftart auch in einer Textform beschrieben werden. Daraus ensteht dann mit Hilfe des Linkers eine kompilierte .FON-Datei.

11.3.2 Der Dialog Editor

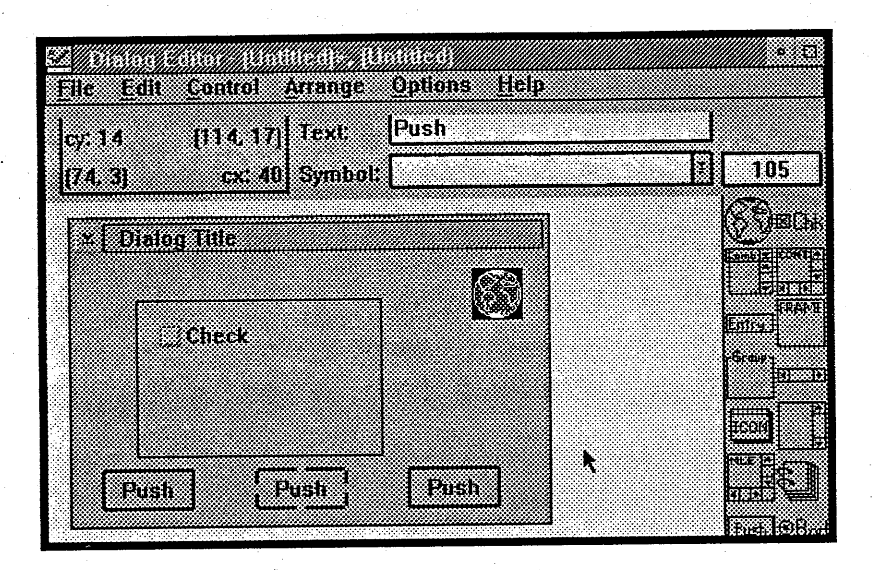


Abbildung 11-4. Dialog Editor (J8HG1103)

Der Dialog Editor ist ebenfalls ein WYSIWYG-Editor. Dialog-Masken werden am Bildschirm erstellt und dann in zwei Formen gespeichert. Die .RES-Dateien sind vom Dialog Box Editor wieder einlesbar und veränderbar. Die .DLG-Dateien sind ASCII-Dateien, welche mit einem Text-Editor weiter bearbeitet werden können.

Die Dialog-Masken sind von den meisten Programmierumgebungen aus verwendbar, nicht jedoch von EASEL, ENFIN/2 oder dem Dialog Manager. Diese Produkte haben eigene Werkzeuge zur Erstellung von Dialogfenstern.

11.4 Informations-Entwicklung

Neben der Programmierung nimmt die Informationsentwicklung eine immer wichtigere Position ein. Daher wurde in OS/2 eine Reihe von Funktionen integriert, welche sowohl die Entwicklung als auch die Darstellung von speziellen Dateien erlaubt. Besonders interessant ist dabei, daß ein Text ohne größere Schwierigkeiten sowohl für **BookMaster** wie auch **OS/2 Online-Dateien** aufbereitet werden kann.

Informationsentwicklung teilt sich in zwei Hauptbereiche:

1. Anwendungsbezogene Information zur Unterstützung bei der Benutzung eines Programmes, d.h. Hilfetexte
2. Eigenständige Information über Vorgänge und Verfahren

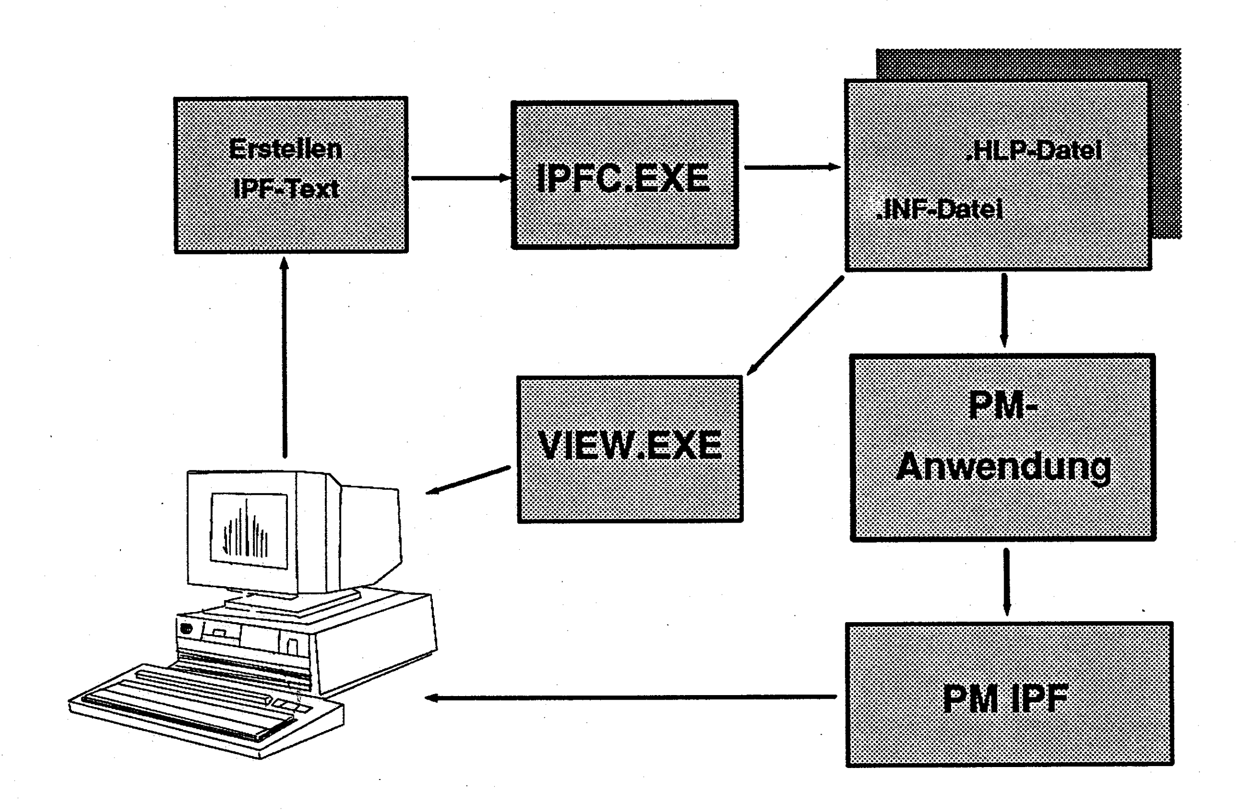


Abbildung 11-5. Konzept des IPF (J8HG1104)

Nach der Art sogenannter Autorensysteme kann Information aufbereitet werden für eine Vielzahl von Anwendungen:

1. Lerntexte mit Grafiken für beliebige Themen, z.B. für Sprachen oder Verwaltungsabläufe
2. OS/2-bezogene Lerntexte, welche zusätzlich zu oben genanntem das Aufrufen von Programmen steuern können
3. Arbeitsanleitungen, z.B. grafische Darstellung eines Montagevorganges in einzelnen Schritten
4. Teilekataloge, Gebäudepläne und andere Art Referenzinformation

11.4.1 Konzept der Informations-Darstellung

Zur **Anzeige** dient das Dienstprogramm VIEW.EXE, welches Teil des Basis-OS/2 ist. Dieses Programm interpretiert den Inhalt von Informationsdateien (mit der Erweiterung INF) nach dem **Hypertext-Verfahren**. Dieses erlaubt das Auffinden von Information nach Indizes, vordefinierten Verzweigungen oder auch nach beliebigen Textstellen. Weiterhin wird Unterstützung geboten wie z.B:

- Kopieren von Texten in eine Datei
- Drucken von Abschnitten
- Verzeichnis bisher angezeigter Seiten
- Blättern zum vorigen Kapitel
- Verwendung von Lesezeichen.

Die anzuzeigenden Texte wurden vom Autor mit Kennzeichen versehen entsprechend ihrer Funktion. Soll ein Begriff als Verzweigungspunkt dienen, so wird er als *link* definiert, was bei der Anzeige zu einer farblichen Absetzung führt. Bei Grafik ist die Einrahmung beliebiger Flächen möglich, welche dann als „Sprungbrett" dienen (*Segmented Hypergrahics*). Der Text ist durch die Farbe als Verzweigungspunkt klar erkennbar; nicht so eine Grafik. Daher sollte bei einer Grafik ein Hinweis auf die Funktion gegeben sein.

11.4.2 Schreiben einer IPF-Datei

Zum Testen der allgemeinen Funktionen von Online-Dokumenten genügt es, wenige Zeilen mit dem Editor zu schreiben. Das folgende Beispiel demonstriert einige der häufigsten Funktionen. Der Dateiname sollte auf **IPF** enden. Das kennzeichnet eine **Eingabe** für den IPFC. Die Ausgaben heißen je nach Verwendungszweck **.HLP** oder **.INF**. Die Texte für beide Dateiarten werden auf gleiche Weise erstellt. Dazu wird nur ein normaler ASCII-Editor gebraucht. Der LPEX (Live Parsing Editor) ist der am besten geeignete Editor, weil er die Möglichkeit einer Text-Vorschau bietet.

Die anzuzeigenden Texte und Grafiken werden für die Verwendung von Hypertext und Hypergrafik strukturiert. Die Struktur der **Online-Dokumente** richtet sich an der Verwendung aus.

1. Dateien als Hilfe zu Programmen (.HLP)
2. Informations-Dateien ohne direkten Programm-Bezug (.INF)

Die Texte werden mit sogenannten IPF-tags versehen. Diese Kennzeichen teilen dem Compiler mit, wie die markierten Texte angezeigt werden sollen und wie sie miteinander verknüpft werden sollen. Fast alle Kennzeichen werden mit einem gleichnamigen Kennzeichen beendet, welches sich durch ein „e“ nach dem Doppelpunkt unterscheidet.

Die Struktur eines On-line-Dokumentes ist zwar weitgehend frei, aber es gibt ein paar Standard-Empfehlungen:

- Texte sollten in ihrem Umfang etwa eine Bildschirmseite haben.
- Texte sollen durch eine Kennzahl (res=nnn) anzuspringen sein.
- Jede Seite hat eine Überschrift

11.4.3 Übung: Verwendung von IPF-Tags

Ziel der Übung

Sie erstellen eine SCRIPT-Datei, wandeln diese mithilfe des IPF-Compilers in eine Informationsdatei um und lassen sich diese mithilfe der VIEW.EXE anzeigen.

Durchführung

1. Als erstes wird die Definition eines Benutzerdokumentes benötigt.

   ```
   :userdoc.
   ```

2. Danach wird die Titelangabe erwartet:

   ```
   :title.Was ist OS/2 Version 2.x?
   ```

3. Und nun das Steuerzeichen, daß der Titel beendet ist:

   ```
   :etitle.
   ```

4. Jetzt kommt die erste Überschrift, der „header“.

   ```
   :h1.Allgemeines
   ```

5. Hier der Inhalt des ersten Fensters „Allgemeines“:

   ```
   Es handelt sich um das zukunftsträchtigste Betriebssystem für
   :color fc=cyan.Personalsysteme:color fc=default.
   ```

6. Wir beschreiben nun das zweite Fenster:

   ```
   :h1 res=100.Hardwarevoraussetzung:
   :ol.
   :li.Prozessor ab 80386
   :li.AT-Bus oder Mikrokanal
   :eol.
   :p.
   :link reftype=hd res=200.Herausragende Merkmale:elink.
   :p.
   :link reftype=hd res=300.Neues in der Version 2.x:elink.
   ```

7. Und nun die Beschreibung des dritten Fensters:

```
:h1 res=200.Herausragende Merkmale
:ol.
:li.Multitasking
:li.Multithreading
:li.Single User
:li.Objektorientiert
:eol.
:p.
:link reftype=hd res=100.Hardwarevoraussetzung:elink.
:p.
:link reftype=hd res=300.Neues in der Version 2.x:elink.
```

8. Die Angaben für das nächste Fenster:

```
:h1 res=300.Neues in der Version 2.x
:ol.
:li.Virtuelle DOS-Maschine
:li.Unterstützung von WINDOWS-Anwendungen
:li.Volle Nutzung der 32bit Adressierung
:eol.
:p.
:link reftype=hd res=100.Hardwarevoraussetzung:elink.
:p.
:link reftype=hd res=200.Herausragende Merkmale:elink.
:euserdoc.
```

So das war's. Speichern Sie diese Datei bitte unter dem Namen

```
HILFE.SCR
```

im Root-Verzeichnis ab. Weiter geht es dann mit der Übung „Erstellen einer Referenzdatei mit IPFC“

11.4.4 Übung: Hilfedatei mit geteilten Fenstern

Ziel der Übung

Sie editieren die erste Datei, oder erstellen diese neu, geben die Parameter an, die für den Aufbau von zwei nebeneinander liegenden Fenstern erforderlich sind und lassen sich das Ergebnis mithilfe der VIEW.EXE anzeigen.

Durchführung

1. Führen Sie diese Übung bitte ohne weitere detaillierte Anweisung durch.

```
:userdoc.
:title.Was ist OS/2 Version 2.x?
:etitle.
```

2. Hier wird das Einstiegsfenster definiert.

```
:h1 res=100 x=left y=bottom width=50% height=100% group=1.
Wählen Sie aus
:p.
:link reftype=hd res=200.Allgemeines
:elink.
:p.
:link reftype=hd res=300.Hardwarevoraussetzung
:elink.
:p.
:link reftype=hd res=400.Herausragende Merkmale
:elink.
:p.
:link reftype=hd res=500.Neues in der Version 2.x
:elink.
```

3. Und nun das erste Sprungfenster.

```
:h1 res=200 x=right y=top width=50% height=100% group=2.
Allgemeines
:p.
Es handelt sich um das zukunftsträchtigste Betriebssystem für
:color fc=cyan.Personalsysteme:color fc=default.
```

4. Hier ein weiteres Sprungfenster.

```
:h1 res=300 x=right y=top width=50% height=100% group=2.
Hardwarevoraussetzung:
:p.
Prozessor ab 80386
:p.
AT-Bus oder Mikrokanal
```

5. Das nächste Sprungfenster.

```
:h1 res=400 x=right y=top width=50% height=100% group=2.
Herausragende Merkmale
:p.
Multitasking, mehrere Anwendungen laufen gleichzeitig
:p.
Multithreading, mehrere threads laufen auf Prozessorebene
nacheinander, für den Benutzer aber gleichzeitig
:p.
Single User, das System ist als Einplatzsystem ausgelegt
:p.
Objektorientiert, der Benutzer sieht seine Anwendung
als Objekt auf dem Bildschirm, er wählt es aus und
führt dann die gewünschte Aktion durch.
```

6. Und noch ein Sprungfenster.

```
:h1 res=500 x=right y=top width=50% height=100% group=2.
Neues in der Version 2.x
:p.
Virtuelle DOS-Maschine
:p.
Unterstützung von WINDOWS-Anwendungen
:p.
Volle Nutzung der 32bit Adressierung
:p.
:euserdoc.
```

So das war's. Speichern Sie das Ergebnis unter dem Dateinamen

FENSTER.SCR

im Root-Verzeichnis ab und führen Sie die Umwandlung durch, wie Sie in der nächsten Übung „Erstellen einer Referenzdatei mit IPFC“ beschrieben ist.

11.4.5 Umwandeln einer Informations-Datei

Das einfachste ist es, eine Befehlszeile aufzumachen und in das Verzeichnis mit dem Quellentext zu wechseln. Z.B.:

```
[C:\]
```

Nun kann der Compiler aufgerufen werden. Die Übung zeigt Ihnen den Aufruf und die Meldungen vom IPFC:

11.4.6 Übung: Erstellen einer Referenzdatei mit IPFC

Ziel der Übung:

- In dieser Übung sollen die Kenntnisse aus „Schreiben einer IPF-Datei" vertieft werden.
- Kennenlernen der Grundlagen von IPF
- Benutzung des IPFC mit einer einfachen BookMaster-Datei
- Anzeigen und Testen der selbsterstellten Informationsdatei

Durchführung

1. Benutzen Sie den Erweiterten Editor zum Schreiben einer kleinen Textdatei mit dem Namen HILFE.SCR oder FENSTER.SCR. welche einige der häufigsten Markierungen wie Listen, Überschriften, Kommentare und Hervorhebungen enthält. Sie finden zwei Muster, die Sie übernehmen können im Kapitel Kapitel 11, „Überblick Anwendungsentwicklung" auf Seite 11-1. Das einfachere Muster stellt einen einfachen Verzweigungstext dar, das andere zeigt die Möglichkeit einen geteilten Bildschirm, *(split window)* zu verwenden.
2. Stellen sie fest, ob der Compiler für IPF-Dateien im Pfad (Toolkt20\IPFC) ist.

   ```
   d:cd\toolkt20\ipfc
   ```
3. Wenn Sie den Compiler ohne Parameter aufrufen, gibt er Ihnen alle Parameter bekannt.
4. Wechseln Sie nun in das Verzeichnis

   ```
   TOOLKT20\IPFC
   ```
5. Rufen Sie nun den Compiler (wieder) auf und geben Sie als Variable den Namen Ihrer Textdatei an und als Parameter /INF:

   ```
   IPFC d:\HILFE.SCR /INF
   ```

   oder

   ```
   IPFC d:\FENSTER.SCR /INF
   ```

   je nachdem welche Datei Sie umwandeln wollen.
6. Der Compiler erstellt Ihnen eine Datei HILFE.INF bzw. FENSTER.INF. Wenn alles geklappt hat, bekommen Sie einen Bericht wie den nachfolgend dargestellten.
7. Bringen sie Ihre INF-Datei mit VIEW.EXE zur Anzeige. Benutzen Sie dazu die Befehlseingabe:

```
VIEW HILFE
```

oder

```
VIEW FENSTER
```

Da Sie sicher objektorientiert arbeiten wollen, legen Sie bitte noch ein Programmobjekt für Ihre selbsterstellte Informationsdatei an. Erinnern Sie sich noch? Es gibt 3 verschiedene Möglichkeiten dafür.

- ein leeres Programmobjekt aus den Schablonen holen
- ein leeres Programmobjekt mit Hilfe des Kontextmenüs eines bestehenden Programmobjektes **Erstellen** und anpassen
- ein bestehendes Programmobjekt kopieren und die Einstellungen anpassen

```
[D:\IPF]ipfc Fenster.scr /inf

Information Presentation Facility Compiler Ver 2.00


(C) Copyright IBM Corporation 1991.  All rights reserved.

Compiling document with following Country Code, Code Page: 49, 850

.....
Resolving cross-references

Cross-references resolved
.....

Statistics :

Document title : Was ist OS/2 Version 2.x?

Total panels created        : 5
Panels defined by name      : 0
Panels defined by number    : 5
Words in dictionary         : 84
Global index entries        : 0
Total index entries         : 0
Total fonts used            : 2
Files referenced externally : 0
Panels defined externally   : 0
Document's Country Code     : 49
Document's Code Page        : 850

Output file successfully created [Fenster.INF]
```

Abbildung 11-6. Kompilierung einer Informations-Datei

11.5 IPF-Tag-Übersicht

Tabelle 11-1. Übersicht IPF-Tags

IPF-Tag	Funktion	Parameter
.*	Kommentar	Beliebiger Text
:artlink.	Leitet die Definition einer Grafik ein, welche als Ziel eines Hypergraphic-Sprunges verwendet wird	
:artwork.	Benennt eine anzuzeigende Grafik-Datei	
:caution.	Warnungstext mit Substitution	
:cgraphic.	Leitet Bereich mit Zeichengrafik ein	
:color.	Farben Text und Hintergrund festlegen	
:dd.	Definition in Liste	
:ddhd.	Überschrift für Definitionen in Liste	
:dl.	Beginn einer Definitions-Liste	
:dt.	Zu definierender Begriff	
:dthd.	Überschrift für Begriffe in Liste	
:fig.	Beginn einer Abbildung	
:figcap.	Kurztext zu Abbildung	
:fn.	Fußnote zum Anzeigen eines Sekundärfensters	

Tabelle 11-2. Übersicht IPF-Tags		
IPF-Tag	**Funktion**	**Parameter**
:h1...h6.	Überschriften Stufen 1 bis 6	
:hide.	Verbergen von Texten, welche nur auf Anforderung gezeigt werden sollen	
:hp1. ... :hp8.	Hervorheben von Texten	
:i1. und :i2.	Indexeinträge	
:icmd.	Identifikation eines Hilfetextes mit Kommando	
:im.	Einfügen einer Datei zur Zeit des Umwandelns	
:isyn.	Synonym für Index definieren	
:li.	*List Item* Eintrag in Liste	
:lines.	Abschalten des Formatierens	
:link.	Verknüpfung zu Text oder Grafik	
:lm.	Linker Rand	
:lp.	Identifizieren einer Erklärung innerhalb einer Liste	
:note.	Definieren eines Kurztextes	
:nt.	Beginn einer Notiz mit mehreren Absätzen	
:ol.	Numerierte Liste	
:p.	Absatz	

Tabelle 11-3. Übersicht IPF-Tags		
IPF-Tag	**Funktion**	**Parameter**
:parml.	Beginn zweispaltige Liste ohne Überschrift	
:pd.	Parameter-Beschreibung	
:.	Parameter	
:rm.	Rechter Rand	
:sl.	Einfache Liste	
:ul.	Unnumerierte Liste	
:userdoc.	Beginn eines Dokumentes	
:warning.	Warnung mit Substitution	
:xmp.	Beispiel ohne Textformatierung	

11.6 Programmierung für OS/2

Es werden drei Arten von Benutzerschnittstellen unterstützt:

1. Zeichenorientierte Schnittstelle,
2. Präsentations-Manager,
3. System Object Model.

11.6.1 Programmierung zeichenorientierter Schnittstellen

In beliebigen Sprachen und ohne sonderliche Schwierigkeiten können Programme geschrieben werden, welche den Gesamtbildschirm benutzen. Das bedarf an dieser Stelle keiner besonderen Erklärung.

11.6.2 Struktur von PM-Programmen

Grundsätzlich sind alle PM-Programme gleich aufgebaut. Sie bestehen aus:

1 Hauptprozedur mit:

- Initialisierung
- Nachrichtenschleife und
- Freigabe von Ressourcen

2 Fenster-Prozedur mit:

- Empfang von Nachrichten
- Untersuchung der Nachricht
- Ausführung aufgrund des Nachrichteninhaltes oder
- Senden einer Nachricht an anderen Dienstgeber
- Default-Prozedur im Falle nicht programmierter Funktion

Alle Nachrichten, ob vom Benutzer oder von Programmen, werden über eine gemeinsame Warteschlange geschickt. Der PM kennt die Adressaten und leitet die Nachrichten an die richtige Anwendung weiter.

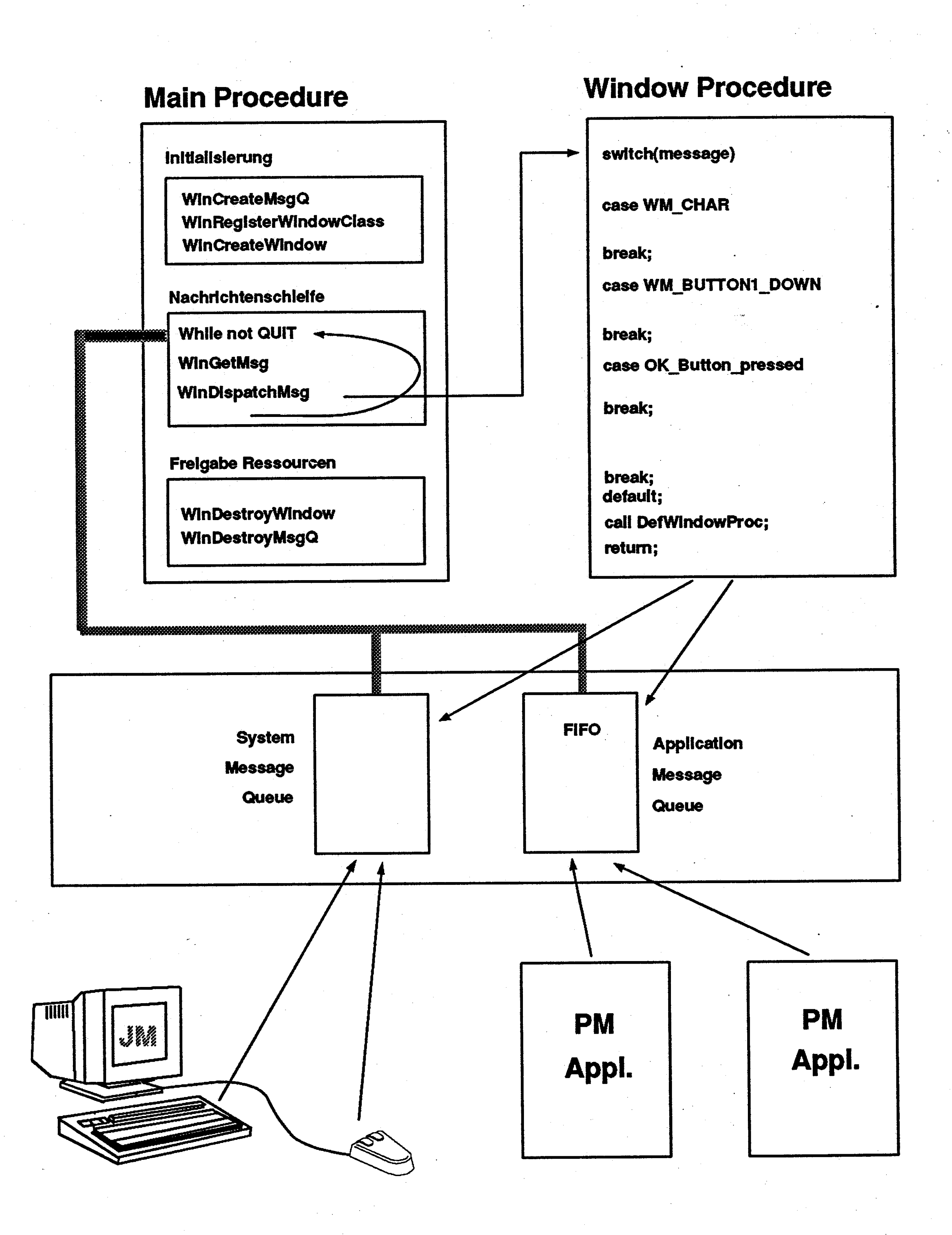


Abbildung 11-7. Konzept der Programme unter PM (J8HG1105)

Das herausragende Merkmal aller PM-Programme ist die Objekt-Orientierte Programmierung (OOP). Sie erfordert bei manchem

Programmierer ein Umdenken von der sequentiellen Programmierung.

Obwohl Sprachen wie C oder COBOL nicht als OOP-Sprachen geschrieben wurden, so lassen sich jedoch mit einiger Disziplin und mit den PM-Erweiterungen Objekte definieren und benutzen.

COBOL/2 unterstützt nicht das Schreiben von Programmen, welche ***reentrant*** sind. Daher können einige PM-Funktionen nicht in COBOL geschrieben werden. Auf dem Umweg über *language bindings* und C-Programme läßt sich einiges erreichen.

11.6.3 Programmieren mit SOM

Die Arbeitsoberfläche des neuen OS/2 wurde mit Hilfe einer neuen Einrichtung, dem *System Object Model*, kurz SOM genannt, geschrieben.

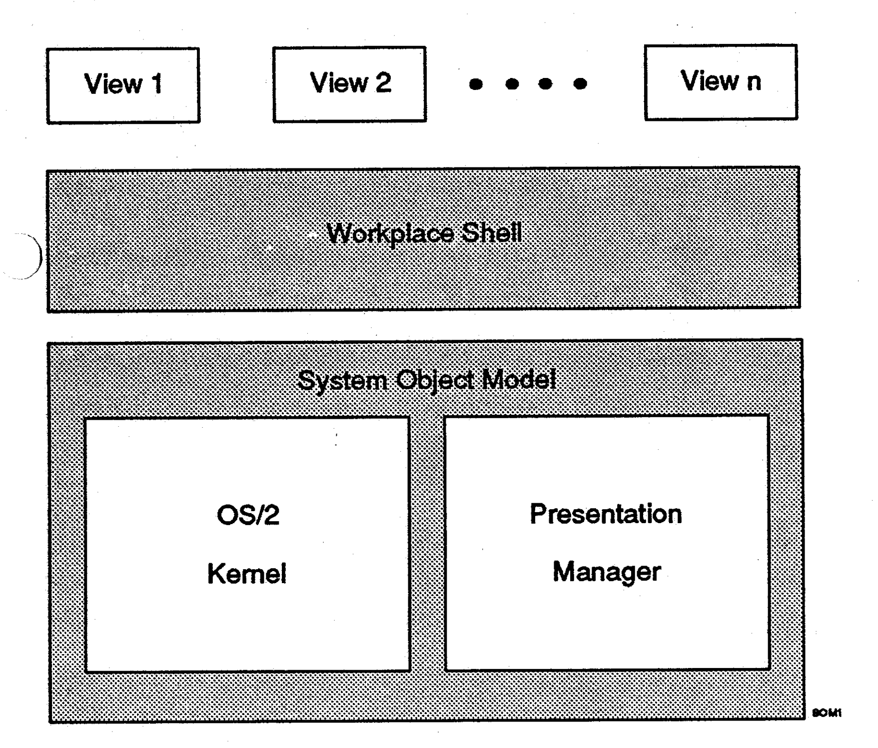


Abbildung 11-8. SOM umfaßt den Kern und den PM (J8HG1108)

Was ist SOM?

- Spezifikation für die Schnittstellen zwischen Objekten
- Language Bindings
- Runtime Facility in OS/2
- Compiler
- Klassen-Hierarchie

Was bietet SOM?

- Portabilität zwischen OO und nicht-OO-Sprachen,
  - Export und Import von Klassen, welche in Sprachen wie C++ oder Smalltalk geschrieben sind,
- Portabilität zwischen Plattformen
  - SOM ist Sprachen-unabhängig
  - SOM ist Plattform-unabhängig

SOM beinhaltet eine definierte Klassenhierarchie.

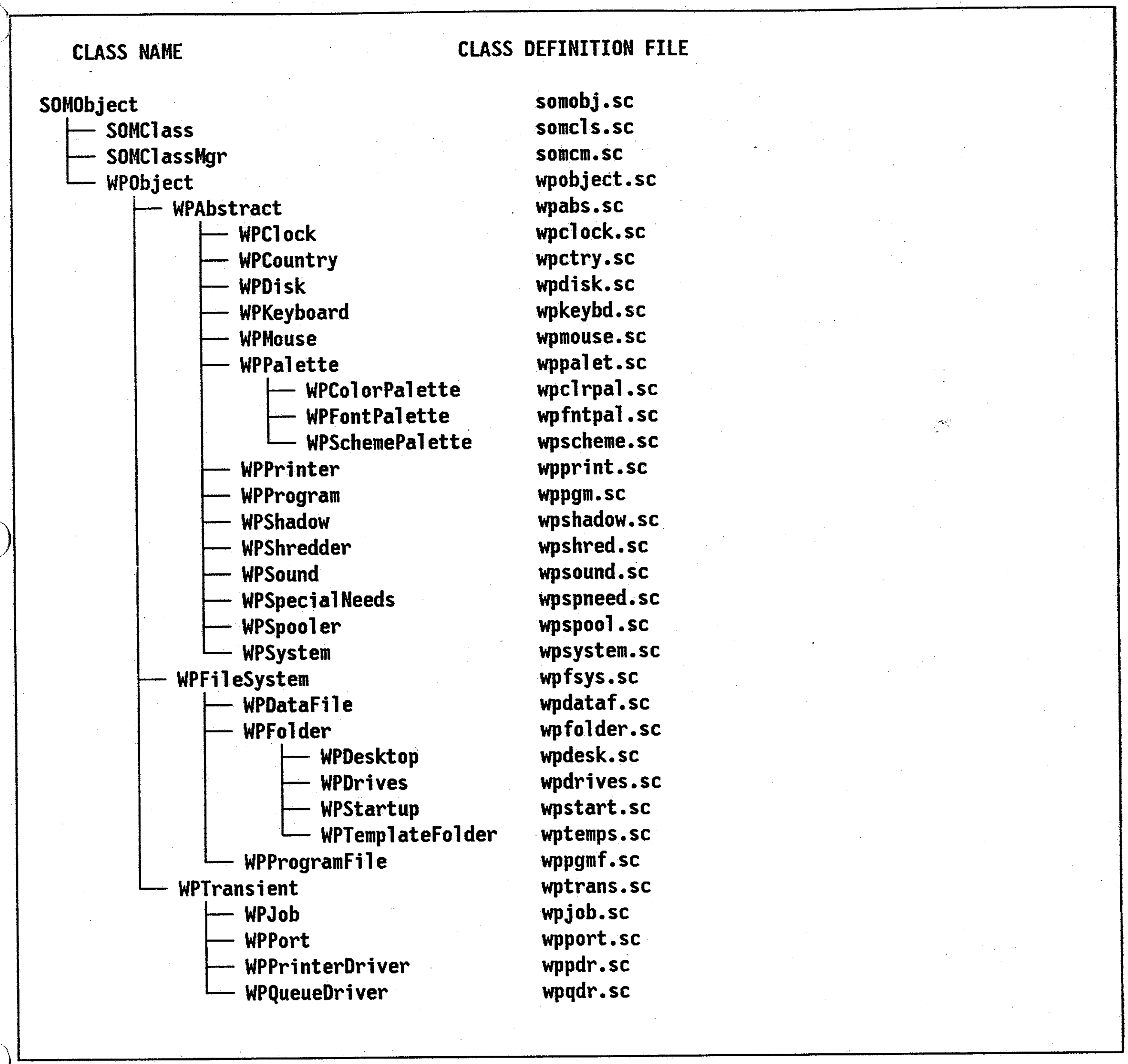
- SOM Klassen haben ihre Methoden in Runtime Facility
- SOM Classes können daher Objekte sein
- SOM hat Meta-Klassen

In SOM werden Objekte in sprachenunabhängig beschrieben und definiert. Das ist möglich durch:

- SOM OIDL - Object Interface Definition Language

Beispiel für SOM

- Die Arbeitsoberfläche ist in SOM geschrieben
- Objekte, welche von der Arbeitsoberfläche benutzt werden, können PM-Objekte oder SOM-Objekte sein.
- Password Folder ist in SOM geschrieben.



```
CLASS NAME                                  CLASS DEFINITION FILE

SOMObject                                         somobj.sc
 ├── SOMClass                                     somcls.sc
 ├── SOMClassMgr                                  somcm.sc
 └── WPObject                                     wpobject.sc
      ├── WPAbstract                              wpabs.sc
      │    ├── WPClock                            wpclock.sc
      │    ├── WPCountry                          wpctry.sc
      │    ├── WPDisk                             wpdisk.sc
      │    ├── WPKeyboard                         wpkeybd.sc
      │    ├── WPMouse                            wpmouse.sc
      │    ├── WPPalette                          wppalet.sc
      │    │    ├── WPColorPalette                wpclrpal.sc
      │    │    ├── WPFontPalette                 wpfntpal.sc
      │    │    └── WPSchemePalette               wpscheme.sc
      │    ├── WPPrinter                          wpprint.sc
      │    ├── WPProgram                          wppgm.sc
      │    ├── WPShadow                           wpshadow.sc
      │    ├── WPShredder                         wpshred.sc
      │    ├── WPSound                            wpsound.sc
      │    ├── WPSpecialNeeds                     wpspneed.sc
      │    ├── WPSpooler                          wpspool.sc
      │    └── WPSystem                           wpsystem.sc
      ├── WPFileSystem                            wpfsys.sc
      │    ├── WPDataFile                         wpdataf.sc
      │    ├── WPFolder                           wpfolder.sc
      │    │    ├── WPDesktop                     wpdesk.sc
      │    │    ├── WPDrives                      wpdrives.sc
      │    │    ├── WPStartup                     wpstart.sc
      │    │    └── WPTemplateFolder              wptemps.sc
      │    └── WPProgramFile                      wppgmf.sc
      └── WPTransient                             wptrans.sc
           ├── WPJob                              wpjob.sc
           ├── WPPort                             wpport.sc
           ├── WPPrinterDriver                    wppdr.sc
           └── WPQueueDriver                      wpqdr.sc
```

Abbildung 11-9. Hierarchie der Workplace Shell Classes

11.7 Unterstützende Programme

Es gibt eine Reihe von Programmen, welche den Programmierern das Arbeiten erleichtern. Dazu gehören:

- REXX - **die** Prozedurensprache für viele kleine Anwendungen und Dienstprogramme
- CODEVIEW - ein Debugger, der einen Programmablauf in Binär- und Quellencode verfolgen und steuern kann
- Programmers Reference des TOOLKIT
- IBM EASEL - eine Programmiersprache in „einfachem Englisch“

Weiterhin gibt es am Markt eine Anzahl Programme, welche bei der Programmierung helfen können. Hier ist jedoch Vorsicht geboten, weil das angebotene Produkt vielleicht nicht dem Stand des OS/2 entspricht.

Da der Präsentations Manager selbst sehr die OOP fördert, bietet sich die Anwendung von ***CASE Tools*** an, welche es von etlichen Herstellern gibt. Einige solche Produkte sind jedoch sehr teuer oder sehr langsam oder auch unvollständig im Leistungsumfang.

11.8 Die Prozedurensprache REXX

11.8.1 Überblick

Die komplexe Systemverwaltung einer Anzahl Maschinen kann es erforderlich machen, daß Prozeduren geschrieben werden müssen, welche über die Möglichkeiten der Stapeldateien hinausgehen. Da sind insbesondere das automatisierte Installieren, Kopieren oder auch Löschen von Dateien zu sehen. Im Rahmen von SAA ist dafür die Sprache REXX vorgesehen, welche eine Vielzahl an Funktionen bietet, welche eine recht einfache Art der Programmierung bietet. Folgende Gruppen von Sprachelementen stehen zur Verfügung:

1. Befehle zur strukturierten Programmablaufsteuerung (ähnlich PL/1)
   - Bedingungen mit if...else
   - Abarbeiten von Blöcken mit do...end
   - Schleifen mit do...Wiederholungsbedingung...end
   - Funktionen und Prozeduren mit Parametern und lokalen Variablen
2. Daten ohne Typenspezifikation
3. Reicher Instruktionssatz zum Untersuchen und Manipulieren von Zeichenketten
4. Hierarchie zum Ausführen verschiedener Kategorien:
   - Benutzer-definierte Prozeduren und Funktionen
   - Prozeduren und Funktionen
   - Vordefinierte Prozeduren und Funktionen
   - Externe Routinen wie:
     - Andere REXX-Programme
     - Betriebssystem-Befehle
     - Kompilierte Programme

11.8.2 Voraussetzungen zur Benutzung von PMREXX

Der REXX-Interpreter ist Teil des OS/2 Standard Edition (OS/2 SE). Er wird bei der Installation des Betriebssystems mitinstalliert. Sie finden ihn und einige dazugehörige Teile im Verzeichnis C:\OS2.

REXX-Programme werden mit einem beliebigen ASCII-Editor geschrieben. Sie können die Namenserweiterung .BAT haben oder .CMD. Die Programmausführung erfolgt unter Steuerung des REXX-Interpreters. Dieser wird automatisch aufgerufen, wenn die erste Zeile einer Stapeldatei ein REXX-Kommentar ist. Der REXX-Interpreter arbeitet gewissermaßen als Erweiterung des Befehlssatzes des Betriebssystems.

Ein Übungsprogramm wird beim TOOLKIT mitgeliefert: REXXTRY. Es erlaubt eine Ausführung etlicher Befehle im Dialog. Damit kann der Programmierer vieles probieren ohne durch die Folge des Editierens, Speicherns und REXX-Aufrufes gehen zu müssen.

Das Programm PMREXX ist ein OS/2-Programm, mit welchem sich REXX-Prozeduren testen und berichtigen lassen.

Beispielaufruf:
PMREXX rexprog[.CMD] [parameter]

11.8.3 Struktur und Syntax

- **Kommentare**
  - **Jedes REXX-Programm muß mit einem Kommentar beginnen.**
  - /* und */ schließen Kommentare beliebiger Länge ein. Kommentare können verschachtelt sein, weshalb auf eine korrekte Paarung der Begrenzer geachtet werden muß. Kommentare können an jeder Stelle stehen, sie wirken trennend, wenn sie innerhalb eines Befehles stehen.
- **Leerstellen**
  - Leerstellen können in beliebiger Zahl an fast jeder Stelle stehen.
  - Sie trennen Befehle, Symbole usw.
- **Zeichenketten**
  - Zeichenketten sind Variable oder Konstante und sind in Apostrophe oder Anführungszeichen eingeschlossen: 'Mainz', "Don't panic"
  - Hexadezimalzahlen sind Zeichenketten, welche Ziffern- und Buchstabenpaare im Bereich 0..9 und A..F enthalten. Sie sind durch ein nachgestelltes x gekennzeichnet: '1f'x, "1d ec f8"X
- **Symbole**
  - REXX benutzt beliebige Zeichenketten als symbolische Namen für Konstanten, Variablen und Schlüsselwörter (Befehle, Funktionen usw).
  - Gültige Zeichen in Symbolen sind: A..Z, a..z, 0..9, #, $, ?, ., _ und Paragraph. Sie werden vor der Benutzung in Großbuchstaben umgewandelt.
  - Ausnahme: In Kombinationen wie 17.3E-12 oder .03e+9 wird das e oder E als Kennzeichen für eine Exponentialzahl erkannt, wenn das Symbol mit einer Ziffer beginnt.

- **Zahlen**
  - Reine Ziffernfolgen sind Zahlen. Sie dürfen . + - und e bzw. E enthalten.
  - Zahlen können Variable oder Konstante sein. Zwischen den Ziffern sind Leerstellen nicht erlaubt.
- **Operatoren**
  - Die Zeichen + - * / % \ & < > ; : not dienen dem arithmetischen oder logischen Verknüpfen von Variablen und/oder Konstanten.
  - Je nach Verfügbarkeit oder Vorzug können das not-Zeichen oder der Schrägstrich rückwärts gleichwertig benutzt werden.
- **Besondere Zeichen**
  - In Verbindung mit anderen Zeichen (außerhalb von Texten) haben **. : ;) (** die Bedeutung von Begrenzungszeichen.

11.8.4 Ausdrücke und Operatoren

REXX lebt von sogenannten Ausdrücken. Sie dienen dem Verknüpfen von beliebigen Daten. Ein Ausdruck enthält mehrere Elemente wie Variable, Konstante und Operatoren, aus denen ein Ergebnis abgeleitet wird.

11.8.5 Anweisungen (Instruktionen)

- ausdruck [;]
  Der Ausdruck wird untersucht und an das Betriebssystem zur Ausführung gegeben.
- name = ausdruck [;]
  Der Ausdruck wird untersucht und der Wert in Name gestellt
- name :
  Eine Benutzerfunktion wird mit Name und Doppelpunkt begonnen
- ARG [maske] [;]
  ARG überträgt eine Zeichenkette, welche einem Programm oder einer Funktion als Argument mitgegeben wurde, in eine oder mehrere Variablen, Dazu kann eine Maske benutzt werden. Häufig wird ARG (gleichbedeutend mit PARSE UPPER ARG) mehrfach hintereinander mit verschiedenen Masken verwendet, um eine Zeichenkette nach verschiedenen Gesichtspunkten zu untersuchen.
- CALL name [ausdruck] [, ausdruck]..... [;]
  Mit CALL werden Routinen aufgerufen. Diese können interne sowie externe Routinen sein oder auch Funktionen.
- DROP namensliste [;]
  Durch DROP werden im weiteren Verlauf eines Programmes die Namen anstelle der Inhalte von den Variablen oder Konstanten benutzt, welche in der Namensliste aufgeführt sind.
- EXIT [ausdruck] [;]
  Bedingungslose Beendigung eines Programmes mit möglicher Übergabe einer Zeichenkette an die aufrufende Routine.
- IF [ausdruck] [;] THEN [;][anweisung] [;]
  [ELSE [;] anweisung] [;]
  Bedingte Ausführung von Anweisungen
- INTERPRET [ausdruck] [;]
  Eine während der Programmausführung zusammengestellte Zeichenkette soll als Anweisung interpretiert und ausgeführt werden.

- ITERATE [name] [;]
  Verändern des Programmablaufs in einer DO-Schleife
- LEAVE [name] [;]
  Verlassen einer oder mehrerer DO-Schleifen
- NOP [;]
  No OPeration, nützlich als Ziel einer Anweisung oder eines Sprunges.
- NUMERIC DIGITS [ausdruck] [;]
  NUMERIC FUZZ [ausdruck] [;]
  NUMERIC FORM [SCIENTIFIC] [;]
  NUMERIC FORM [ENGINEERING] [;]
  NUMERIC FORM [[VALUE] ausdruck] [;]
  DIGITS steuert die Genauigkeit einer Zahl, FUZZ beschränkt die Genaugkeit vorübergehend, um Vergleiche zu erleichtern. FORM legt die Darstellungsweise einer Zahl fest.
- OPTIONS [ausdruck] [;]
  Diese Anweisung leitet besondere Anforderungen an REXX weiter.
- PARSE [UPPER] SOURCE maske [;]
  PARSE [UPPER] VERSION maske [;]
  PARSE [UPPER] VALUE ausdruck WITH maske [;]
  PARSE [UPPER] VAR name maske [;]
  PARSE [UPPER] ARG maske [;]
  PARSE [UPPER] PULL maske [;]
  PARSE [UPPER] LINEIN maske [;]
  Daten von verschiedenen Quellen werden Variablen nach verschiedenen Verfahren zugewiesen. ARG zeigt auf Daten, welche als Argument übergeben wurden, LINEIN liest die nächste Zeile, PULL das nächste Element einer Warteschlange, VERSION holt die Version von REXX, VALUE untersucht einen Ausdruck und VAR veranlaßt REXX, den Inhalt einer Variablen zu zerlegen.
- PROCEDURE [EXPOSE namensliste] [;]
  Mit dieser Anweisung lassen sich die benannten Variablen vor den folgenden Anweisungen verbergen.

- PULL [maske] [;]
  Liest Daten von einer Daten-Warteschlange.

- PUSH [ausdruck] [;]
  Schiebt Daten in eine Daten-Warteschlange; LIFO.

- QUEUE [ausdruck] [;]
  Schiebt das Ergebnis des Ausdruckes in die Warteschlange

- RETURN [ausdruck] [;]
  Kehrt zum Ausgangspunkt zurück.

- SAY [ausdruck] [;]
  Das Ergebnis des Ausdruckes wird an Standard-Ausgabe gegeben,

- SELECT [;] whenlist [OTHERWISE [;] [anweisungsliste]] END [;]
  Wählt eine von mehreren alternativen Anweisungen aus.

- SIGNAL labelname [;]
  SIGNAL [VALUE] ausdruck [;]
  SIGNAL [ON ' OFF] bedingung [;]
  Der Ablauf eines Programmes wird eingestellt und eine Ausnahmebedingung 'signalisiert'. Die Aktion hängt von der Verwendung ab.

- TRACE [parameter] [;]
  TRACE [[VALUE] ausdruck] [;]
  Mehrere Parameter können angegeben werden, um festzulegen, was mit TRACE verfolgt werden soll

11.8.6 Standard-Routinen

- ABBREV
- ABS (zahl)
- ADRESS()
- ARG([argno [, option]])
- CENTER(zeichenkette, k [,pad]) / CENTRE(zeichenkette, k [,pad])
- COMPARE(zeichenkette1, zeichenkette2 [, pad]
- COPIES(zeichenkette, n)
- DATATYPE(zeichenkette [, type])
- DATE ([option])
- DELSTR(zeichenkette, n [, länge])
- DELWORD(zeichenkette, n [, länge])
- DIGITS()
- FORM()
- FORMAT(zahl [, [vorher] [, [nachher]]])
- FORMAT(zahl [, [vorher] [, [nachher] [expp] [, expt]]])
- FUZZ()
- INSERT(neu, ziel [, [n] [, [länge] [, pad]]])
- LASTPOS(zeichenkette1, zeichenkette2 [, start])
- LEFT(zeichenkette, k [,pad])
- LENGTH(zeichenkette)
- LINEIN([name] [, [zeile] [, zähler]])
- LINEOUT([name] [, [zeichenkette] [, zeile]])
- LINES([name])
- MAX(zahl , [zahl] . . .)
- MIN(zahl , [zahl] . . .)
- OVERLAY(neu , ziel [, [n] [länge] [, pad]]])
- POS(zeichenkette1, zeichenkette2 [, start])
- QUEUED()
- RANDOM([min] [, [max]] [, saat])

- REVERSE(zeichenkette)
- RIGHT(zeichenkette, k [,pad])
- SIGN(zahl)
- SPACE(zeichenkette [, [n] [,pad]])
- STRIP(zeichenkette [, [option] [,char]])
- SUBSTR(zeichenkette , pos [, länge , pad]])
- SUBWORD(zeichenkette , n [,k])
- SYMBOL(name)
- TIME([option])
- TRACE([parameter])
- TRANSLATE(zeichenkette [, [tableo] [, [tablei] [, pad]]])
- TRUNC(zahl [, n])
- VALUE(name)
- VERIFY(zeichenkette, vergleichsliste [, [option] [start]])
- WORD(zeichenkette , n)
- WORDINDEX(zeichenkette , n)
- WORDLENGTH(zeichenkette , n)
- WORDPOS(satz , zeichenkette [, start])
- WORDS(zeichenkette)
- ASK(['noecho'])
- VER()
- GETENV(zeichenkette)
- DIRECTORY([neuesverzeichnis])
- READ(dateiname , [, position])
- SEEK(dateiname , [, position])
- SIZE(dateiname)
- WRITE(dateiname , zeichenkette [, option])
- FINISH(dateiname)

Anhang A. Anhang spezielle Übungen

Übung

Planung einer komplexen Installation

Ziel der Übung:

In diesem Abschnitt üben Sie die Planung der Installation, unter Berücksichtigung verschiedener Kriterien.

1. Ergebnis soll eine Partitionierungsübersicht für verschiedene Zielgruppen sein.
2. Als Ergebnis soll pro Arbeitsgruppe 1 Partitionierungstabelle vorliegen, welche den anderen Arbeitsgruppen vorgestellt und erläutert wird. Wichtig ist hierbei die Auseinandersetzung mit den Installationskriterien, eine Musterkonfiguration für Ihren Arbeitsplatz kann sicher ein anderes Aussehen haben.

Durchführung

1. Bearbeiten Sie den für Ihre Arbeitsgruppe (z.B. Tischreihe) vorgesehenen Arbeitsplatz.
2. Ergänzen Sie das Bedarfsprofil für den zu bearbeitenden Arbeitsplatz gemäß Ihren Erfahrungen oder Erwartungen.
3. Berücksichtigen Sie bei der Planung folgende Kriterien:
   - Anwendungen
     - ____________________
     - ____________________
   - Betriebssysteme
     - ____________________
     - ____________________
   - Hardware
     - ____________________

— ____

- Dateiverwaltungssystem

  — ____

  — ____

- Partitionierung

  — ____

  — ____

Erläuern Sie ggf. Ihre Entscheidungen.

- ____
- ____

4. Fassen Sie die gewählte Partitionierung nach folgenden Kriterien zusammen:

   - Art der Partition (primär/erweitert)
   - Verwendungszweck (System/Anwendung)
   - Dateiverwaltungssystem
   - Laufwerksbuchstabe
   - Platzbedarf (geschätzt)

5. Nachdem alle Arbeitsgruppen ihr Ergebnis erarbeitet haben, stellen Sie es allen Kursteilnehmern vor. Die Wahl der Mittel dafür ist Ihnen freigestellt.

Hier nun die Aufgabe für die Gruppe 1:

1. Sekretariats-/Sachbearbeiterarbeitsplatz (Einzelplatzsystem) Hier werden i. d. R. Standardprodukte für Textverarbeitung, Tabellenkalkulation, Datenbank und evtl. noch einfache Graphikerstellung genutzt.
2. Hier Ihre Vorgaben:
   - Festplatte: 120 MB
   - Anwendungen: Textverarb., Tabellenkalk., Comm. Manager
   - Sie wissen, daß ein DOS-Programm aus eigenem Hause zu erwarten ist. Es ist Ihnen nicht bekannt, ob dieses *„ill behaved“* ist.
   - ______________________________
   - ______________________________

Hier die Aufgabenstellung für die Gruppe 2:

1. Sachbearbeiterplatz mit LAN-Anbindung.
2. Hier werden i. d. R. Standardprodukte für Textverarbeitung, Tabellenkalkulation und Datenbank genutzt, diese sind meist auf Netzwerklaufwerke ausgelagert.
3. Hier Ihre Vorgaben:
   - Festplatte: 60 MB
   - Anwendungen: Textverarb., Tabellenkalk.
   - ______________________________
   - ______________________________

Die Aufgabenstellung für die Arbeitsgruppe 3:

1. Sekretariats-/Sachbearbeiterarbeitsplatz (mit HOST-Anschluß)
2. Hier werden i. d. R. Standardprodukte für Textverarbeitung, Tabellenkalkulation, Datenbank und evtl. noch einfache Graphikerstellung genutzt, Zugang zum Großrechner.
3. Ihre Vorgaben sind:
   - Festplatte: 200 MB
   - Anwendungen: Textverarb., Tabellenkalk., OS/2 Ext. Services
   - Sie wissen, daß ein DOS-Programm aus eigenem Hause zu erwarten ist. Es ist Ihnen nicht bekannt, ob dieses *„ill behaved"* ist.
   - ______________________________
   - ______________________________

Die Aufgabenstellung für die Arbeitsgruppe 4:

1. Benutzerservice-Arbeitsplatz
2. Hier werden i. d. R. nicht nur Standardprodukte eingesetzt. Zur Unterstützung der verschiedenen Benutzergruppen, müssen deren Arbeitsplätze nachgebildet werden können.
3. Hier Ihre Vorgaben:
   - Festplatte: 400 MB
   - Anwendungen: div. Betriebsysteme, Standard-SW, Analysetools
   - ______________________________
   - ______________________________

Hier die Aufgabenstellung für die Gruppe 5:

1. Grafiker-Arbeitsplatz. Hier werden i. d. R. Standardprodukte eingesetzt, Programme wie auch die zu speichernden Dateien benötigen sehr viel Speicherplatz.
2. Hier Ihre Vorgaben:
   - Festplatte: 600 MB (Min.)
   - Anwendungen: CAD- und DTP-Programme
   - Sie wissen, daß ein DOS-Programm aus eigenem Hause zu erwarten ist. Es ist Ihnen nicht bekannt, ob dieses *„ill behaved“* ist.
   - ______________________________
   - ______________________________

Hier die Aufgabenstellung für die Gruppe 6:

1. Programmierer-Arbeitsplatz
2. Hier werden nicht nur Standardprodukte eingesetzt, die Programmentwicklung muß evtl. unter verschiedenen Systemumgebungen stattfinden können.
3. Hier Ihre Vorgaben:
   - Festplatte: 400 MB
   - Anwendungen: div. Betriebsysteme (u.a.: DOS 3.3), Standard-SW, Entwicklungs- und Analysetools
   - ______________________________
   - ______________________________

Hier der Vorschlag, wie Ihre Partionierungstabelle aufgeteilt sein könnte.

Part.	Verwendungszweck	Filesyst.	LW	Platz
_	_	_	_	_
_	_	_	_	_
_	_	_	_	_

Ein paar Hinweise für die Praxis:

- Im Installationshandbuch finden Sie viele nützliche Hinweise zur Installationsplanung wie z.B. DOS-Voraussetzungen (Primäre Partition, maximale Partitionsgröße usw.) oder die Installation mit einer Antwortdatei.
- Ausgehend von einer bestehenden Installation, können Sie zusätzlich noch die Online-Hilfe und Utilities wie FDISK oder FDISKPM nutzen.

Übung:

Schreiben einer Stapeldatei mit Variablen

Ziel der Übung

Es soll eine Stapeldatei geschrieben werden, die es ermöglicht auf Laufwerk A: eine Diskette mit 720 kB und auf Laufwerk B: eine Diskette mit 1,44 MB zu formatieren.

Durchführung

Im ersten Schritt geben Sie ein:

```
@echo off
```

Dies hat zur Folge, daß beim Laufen der Datei keine Ablauf- Kommandos zu sehen sind.

Nun geben Sie die erste "if" -Abfrage ein:

```
if %1 == a: goto 720
```

Nun drei weitere...

```
if %1 == A: goto 720
if %1 == b: goto 1440
if %1 == B: goto 1440
```

Mit Doppelpunkt werden die Verzweigungspunkte definiert

```
:720
```

Nun kommt die durchzuführende Aktion

```
format %1 /n:9 /t:80
```

Da die Aktion nun beendet ist, erfolgt die Verzweigung zum Ende der Stapeldatei

```
goto end
```

hier nun der zweite Verzweigungspunkt:

```
:1440
format %1 /F:1440
```

und nun der letzte Verzweigungspunkt:

:end

voila !

Übung:

Schreiben einer Stapeldatei mit Variablen

Ziel der Übung

Es soll eine Stapeldatei geschrieben werden, die nur das Formatieren von Disketten erlaubt, von Festplatten aber verhindert.

Durchführung

Im ersten Schritt geben Sie ein:

```
@echo off
```

Dies hat zur Folge, daß beim Laufen der Datei keine Ablauf- Kommandos zu sehen sind.

Sie können Ihrer Stapeldatei Kommentare/Erklärungen zufügen:

```
REM Sicheres Formatieren von Disketten
REM Diese Datei bekommt den Namen
REM        FORMAT.CMD
REM und steht im Verzeichnis \OS2
REM Voraussetzung dafür ist, daß vorher
REM die Datei FORMAT.COM in _FORMAT.COM
REM umbenannt wird, z.B. mit folgendem
REM Befehl:
REM ren \os2\format.com \os2\_format.com
```

Nun geben Sie die erste "if" -Abfrage ein:

```
if %1.==. goto GetDrv
```

damit wird gesteuert, daß bei Angabe „format" ohne weitere Parameter, sofort zu GetDrv verzweigt wird

```
for %%d in (A:,a:,B:,b:) do if %1==%%d goto FR
```

hier wird eine "FOR - DO Schleife" gebildet, die die 4 möglichen Diskettenangaben enthält. Bei Zutreffen eines dort definierten Parameters wird nach FR verzweigt

Trifft keine der genannten Bedingungen zu läuft die Stapeldatei zur nächsten Zeile:

```
goto NoFormat
```

es wird nach NoFormat verzweigt

hier nun der erste Verzweigungspunkt:

```
:FR
_format %1
goto end
```

und hier der zweite Verzweigungspunkt:

```
:NoFormat
if %1 == $ _format %2
```

hoppla, was ist das denn?

Mit diesem Trick kann der „Wissende“ dann doch wieder eine Festplatte formatieren und zwar z.B. mit der Eingabe: format $ c:

```
if %1 == $ goto end
```

hier wird wieder zum Ende verzweigt

```
echo Nicht die Festplatte formatieren
```

Dieser Hinweis erfolgt, wenn versucht wurde eine Festplatte (ohne den Trick mit „$“) zu formatieren Da die Aktion nun beendet ist, erfolgt die und hier wieder ein Verzweigungspunkt:

```
:GetDrv
echo Bitte Befehl neu eingeben
echo Beispiel: FORMAT A:
```

und nun der letzte Verzweigungspunkt:

```
:end
```

!!! geschafft !!!

A.2 Der Symboleditor

Der Symboleditor dient dem Erstellen von Piktogrammen, Zeigern oder kleinen Rastergrafiken. Diese drei Datentypen unterscheiden sich nur in der Art ihrer Benutzung. Dafür haben sie ganz bestimmte Eigenschaften.

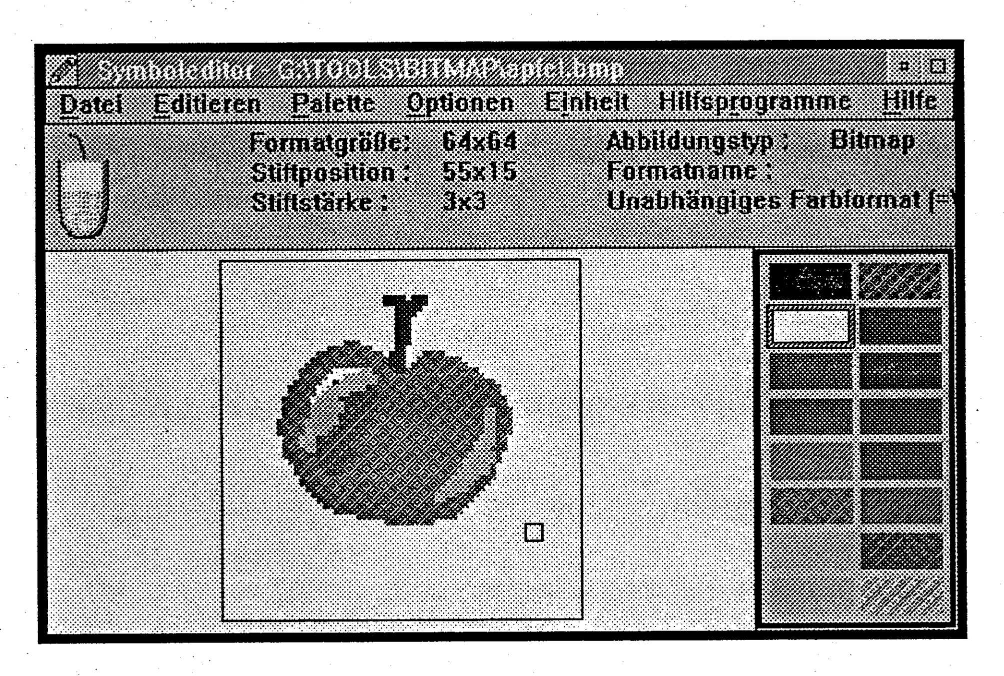


Abbildung A-1. Hauptfenster Symboleditor (J8HG0910)

Der Symboleditor kann beliebige bitmap-Grafiken verwenden. Das ist jedoch nicht sein Zweck. Zum Malen und Zeichnen von allgemeiner Grafik sollte ein ensprechendes Anwendungsprogramm benutzt werden.

Die Piktogramme lassen sich je nach zu verwendender hardware in unterschiedlichen Formaten (VGA, 8514A usw) speichern. Je nach Festlegung, wie ein *ICON* (oder *bitmap* oder *pointer*) gemacht werden soll, kann es sein, daß das Ergebnis bei der Anzeige nicht ganz dem entspricht, was in der Musteranzeige des Symboleditors zu sehen war.

A.3 Weitere Dienstprogramme

Zu Ihrer Information haben wir hier für andere Dienstprogramme aus dem Ordner **Produktivität** einfache Übungen definiert. Diese sind selbsterklärend geschrieben worden.

Viel Spaß und Erfolg damit.

A.3.1 Übung: Arbeiten mit dem Symboleditor

Ziel der Übung

Sie erstellen ein eigenes Symbol, und ordnen es einer Datei zu.

Durchführung

1. Öffnen Sie den Ordner Produktivität
2. Aktivieren Sie den Symboleditor
3. Starten Sie mit Datei, Neu...
4. Wählen Sie aus der Farbpalette auf der rechten Seite mit der ″linken″ Maustaste die Farbe aus, die Sie auf der Taste haben möchten, tun Sie das gleiche mit der ″rechten″ Maustaste.

Nun lassen Sie Ihrer Phantasie freien Lauf. Wenn Sie andere Farben auf den Maustasten haben wollen, wählen Sie neu aus. Sollten Ihnen die Standardfarben nicht ausreichen, wählen Sie erst die Farbe aus, die Sie verändern möchten. Wählen Sie dann Palette und Farben editieren... Zu guter letzt speichern Sie Ihr Symbol mit eigenem Namen und der Erweiterung .ICO ab. (Damit Sie es auch leicht wiederfinden, schlage ich vor, im Root-Verzeichnis.
Nun schließen Sie den Symboleditor mit Doppelklick in der linken oberen Ecke.
Greifen Sie nun die Datei BUTOEXEC.BAT. Gehen Sie in das Kontextmenü (″rechte″ Maustaste) und innerhalb dem Notizbuch,

das Sie über den Begriff Einstellungen öffnen, wählen Sie die Seite Allgemein
Im nächsten Schritt klicken Sie die Taste Suchen an, den Namen können Sie bereits eingrenzen, z.B. *.ico.
Klicken Sie Lokalisieren an und danach z.B. die Zunge Pfad.
Geben Sie hier den Pfad an, klicken Sie OK und dann Suchen an. Klicken Sie Ihr Symbol an und wenn es in dem Notizbuch Ihrer BUTOEXEC.BAT erscheint, schließen Sie es wieder mit Doppelklick.

So das war's. Ich hoffe, es gefällt Ihnen.

A.3.2 ÜBUNG: Die Datenbank

Ziel der Übung

Sie erfahren, wie Sie sich ganz leicht und schnell eine kleine Datenbank erstellen können

Durchführung

1. Wir beginnen indem wir den Ordner Produktivität öffnen.
2. Aktivieren Sie das Objekt Datenbank mittels Doppelklick
3. Beginnen Sie mit der Auswahl Datei, Neu
4. Nun Öffnen Sie den Menüpunkt Editieren und dann Kopfzeilen editieren
5. Jetzt hat sich in Ihrem Datenbankblatt auf der linken Seite ein weißer Eingabebereich geöffnet. Machen Sie Eingaben, z.B. Name, Vorname, Tel-Nr., Straße, PLZ, Ort als Beschreibung für die später einzugebenden Datenzeilen
6. Wenn Sie damit fertig sind, klicken Sie wieder Datei und dann Sichern an in dem sich öffnenden Dialogfenster geben Sie Ihrer neuen Datenbank einen Namen und wählen aus Sichern
7. Sie sehen, Ihre Angaben sind nun wirksam und Ihre Datenbank trägt den von Ihnen erteilten Namen
8. Zur Eingabe Ihre ersten Datensatzes beginnen Sie mit Editieren und Neuen Datensatz hinzufügen
9. Wichtig: nach der Eingabe jedes Datensatzes müssen Sie Datei und Sichern wählen
10. Danach können Sie mit:Editieren und Neuen Datensatz hinzufügen einen weiteren Datensatz anlegen. Vergessen Sie aber bitte nicht das jeweilige Sichern.
11. Nachdem Sie einige Datensätze angelegt haben, klicken Sie bitte den Menüpunkt Anzeigen an. Sie definieren hier den Auswahlschlüssel

12. Nun sehen Sie unter dem Feld Suchschlüssel die Suchschlüsseldaten aller Ihrer Datensätze

13. Wenn Ihnen die Farben und/oder die Schriftgröße nicht zusagen, klicken Sie bitte Anpassen und suchen Sie aus, was Ihnen zusagt.

A.3.3 Übung: Das Programm Haftnotiz

Ziel der Übung

Sie lernen die Möglichkeiten kennen, sich Haftnotizen auf den Bildschirm zu legen.

Durchführung

Stellen Sie sich das Objekt vor als einen Stapel von 10 Klebezetteln. Von diesem können Sie sich einen oder mehrere auf den Bildschirm „kleben".

1. Aktivieren Sie das Objekt Haftnotiz und ziehen Sie sich den ersten Zettel, vom Stapel.
2. Machen Sie einige Texteingabe, Sie werden feststellen, daß bei dem ersten Zeichen automatisch Datum und Uhrzeit eingeblendet werden.
3. Mit Grafiken können Sie sich eines der dort dargestellten Symbole in Ihren Text integrieren.
4. Mit dem Menüpunkt Anpassen und Farben... können Sie sich Farben auswählen.
5. Schriftartgröße ermöglicht, die Schriften zwischen 10x6 und 22x12 zu variieren, wobei auch die Haftnotizgröße sich anpasst.
6. Mit Symbol können Sie definieren wo die Haftnotiz auf Ihrem Bildschirm stehen soll.
7. Klicken Sie den Minimize Button des Objektes Haftnotiz an.
8. Führen Sie einen Systemabschluß durch und starten Sie das System wieder.
9. Wenn alles geklappt hat, stehen Ihre Haftnotizen an der definierten Stelle am Bildschirmrand.

A.3.4 Übung: Das Programm Tabellenkalkulation

Ziel der Übung

Sie lernen die Möglichkeiten kennen, diese Anwendung für kleine Tabellen zu nutzen.

Durchführung

1. Beginnen Sie wie immer mit Datei Neu.
2. Gehen Sie nun mit dem Cursor auf die Zelle A1 und geben Sie eine beliebige Zahl ein.
3. Nun tun Sie das gleiche in der Zelle A2.
4. Und noch ein drittes Mal in der Zelle A3.
5. Um das Ergegnis optisch ein wenig abzugrenzen, gehen Sie nunmehr zwei Zellen tiefer in die Zelle A5.
6. Unterhalb des Eingabefeldes Wert sehen Sie das Feld Formel. Geben Sie hier bitte ein:

   ```
   A1+A2+A3
   ```

   und drücken Sie ENTER.
7. Nun geben Sie bitte in die Zellen B1 bis B3 ebenfalls Zahlen ein.
8. In die Zelle B5 im Feld Formel geben wir jetzt eine Summenformel ein:

   ```
   B1@B3
   ```

9. Sofern Sie unter Berechnen den Menüpunkt Automatisches Berechnen aktiviert haben, werden nun bei jeder Änderung sofort die Summen korrigiert, wie sich das für ein anständiges Tabellenkalkulationsblatt gehört.

Anhang B. Abkürzungen aus dem US-Englischen

Abbreviation	Explanation
ABEND	ABnormal END of task.
ABI	Application Binary Interface
ABIOS	Advanced Basic Input/Output System (IBM)
ACDI	Asynchronous Communication Device Interface (IBM)
Ack	Acknowledge
ADF	Adapter Description File (IBM)
ADP	Adapter Description Program (IBM)
AEA	Asynchronous Emulation Adapter (IBM)
AFP	Advanced Function Printing (IBM)
AI	Artificial Intelligence
AIX	Advanced Interactive eXecutive (IBM)
ALGOL	ALGOrithmic Language
ALU	Arithmetic and Logic Unit
ANSI	American National Standards Institute
APAR	Authorized Program Analysis Report
APDU	Application Protocol Data Unit (OSI)
API	Application Programming Interface
APL	A Programming Language
APPC	Advanced Program-to-Program Communications (IBM)
APPN	Advanced Program-to-Program Networking (IBM)
ARP	Address Resolution Protocol (TCP/IP)
ARTIC	A RealTime Interface Coprocessor (IBM)
ASC	ASynchronous Communications
ASCII	American national Standard Code for Information Interchange (ANSI)
ASD	Application Solutions Division (IBM)
AT&T	American Telephone & Telegraph
AVC	Audio-Visual Connection (IBM)
BASIC	Beginner's All-purpose Symbolic Instruction Code programming language
BBS	Bulletin Board System
BICARSA	Billing, Inventory Control, Accounts Receivable, Sales Analysis
BIOS	Basic Input/Output System
bit	BInary digiT
BitBLT	Bit BLock Transfer
BMP	Bitmap
BSC	Binary Synchronous Communications (= Bisynchronous = Bisync)
BTU	Basic Transmission Unit (SNA)
BTU	British Thermal Unit
CAD	Computer-Aided Design
CAD/CAM	Computer-Aided Design/Computer-Aided Manufacturing
CAE	Computer-Aided Engineering
CAI	Computer-Assisted Instruction
CASE	Computer-Aided Software Engineering
CAT	Computer-Assisted Training
CBEMA	Computer and Business Equipment Manufacturers Association
CBT	Computer-Based Training
CCCCP	Committee on Computer-to-Computer Communication Protocols
CCITT	Comite Consultatif International de Telegraphic et Telephonie
CD ROM	Compact Disk Read-Only Memory
CEN	Comite Europeen de Normalisation
CENELEC	Comite Europeen de Normalisation ELECtrotechnique
CEPT	Conference Europeenne des administrations des Postes et des

	Telecommunications
CGA	Color Graphics Adapter (IBM)
CICS	Customer Information and Control System (IBM)
CIM	Computer-Integrated Manufacturing
CISC	Complex Instruction-Set Computer
CM	Communication Manager (IBM OS/2 EE)
CMAS	Construction Management Accounting System (IBM)
CMC	Communications Management Configuration (IBM)
CMIP	Common Management Information Protocol
CMIS(E)	Common Management Information Service (Element)
CMOT	Centrally Managed OSI protocol for TCP/IP
CMP	Communication Management Processor
CNM	Communication Network Management
COBOL	COmmon Business-Oriented Language
CoCom	COordinating COMmittee for Multilateral Export Control
CODASYL	COnference on DAta SYstems Languages
Comecon	COuncil for Mutual ECONomic Assistance (Eastern Bloc)
COMMON	IBM 1130, 1620, 1800, S/360 User Group
COS	Corporation for Open Systems
CPI	Common Physical Interface (ISDN)
CPI-C	Common Programming Interface for Communications (IBM)
CPS	Characters Per Second
CPU	Central Processing Unit
CRC	Cyclic Redundancy Check
CRT	Cathode Ray Tube
CSA	Canadian Standards Association
CSD	Corrective Service Diskette
CSD	Customer Service Division (now National Service Division) (IBM)
CSMA/CA	Carrier Sense, Multiple Access with Collision Avoidance
CSMA/CD	Carrier Sense, Multiple Access with Collision Detection (ISO 802.3)
CTS	Clear To Send
CUA	Common User Access
CUT	Control Unit Terminal (IBM)
CXI	Common X Interface
DAC	Data Acquisition Controller (IBM)
DAC	Digital-to-Analog Converter
DAP	Developer Assistance Program (= IDAP)
DARPA	Defense Advanced Research Projects Agency
dB	Decibel
DBM	DataBase Manager (IBM OS/2 EE)
DBMS	DataBase Management System
DB2	DataBase 2 (IBM)
DCA	Document Content Architecture
DCE	Data Circuit terminating Equipment (CCITT)
DES	Data Encryption Standard (NIST)
DFT	Distributed Function Terminal (IBM)
DIA	Document Interchange Architecture (IBM)
DIP	Dual Inline Package
DIS	Draft International Standard
DM	Dialog Manager
DMAS	Distribution Management Accounting System
DNS	Domain Name Service (TCP/IP)
DOS	Disk Operating System (IBM/Microsoft)
DPI	Dots Per Inch
DSO	Dealer Service Option (IBM)

DTE	Data Terminal Equipment (CCITT)
DVI	Digital Video Interactive (Intel/IBM)
EBCDIC	Extended Binary-Coded Decimal Interchange Code (IBM)
ECF	Enhanced Connectivity Facility (IBM)
ECMA	European Computer Manufacturers Association
EDI	Electronic Data Interchange
EDIFACT	EDI For Administration, Commerce, and Transportation
EEPROM	Electronically-Erasable Programmable Read-Only Memory
EGA	Enhanced Graphics Adapter (IBM)
EGP	Exterior Gateway Protocol (TCP/IP)
EISA	Extended Industry Standard Architecture
EMS	Expanded Memory Specification (Lotus/Intel/Microsoft)
EPROM	Erasable Programmable Read-Only Memory
ESDI	Enhanced Small-systems Device Interface
ESF	Enhanced Spooling Facility (Product for IBM mainframes)
ETSI	European Telecommunications Standards Institute
EWOS	European Workshop on Open Systems
FAT	File Allocation Table (IBM/Microsoft)
FCBS	File Control BlockS
FCC	Federal Communications Commission
FDDI	Fiber Distributed Data Interface (ISO)
FIPS	Federal Information Processing Standard
FORTRAN	FORmula TRANslation programing language
FRU	Field Replaceable Unit
FTAM	File Transfer, Access, and Management (ISO)
FTP	File Transfer Protocol (TCP/IP)
FTTERM	PC/Host File Transfer and TERMinal Emulator Program (IBM)
FTZ	Fernmeldetechnisches Zentralamt
Gb	Gigabit (= 1,024Mb)
GB	Gigabyte (= 1,024MB)
GDDM	Graphical Data Display Manager (IBM)
GGP	Gateway-to-Gateway Protocol
GOP	General Operating Permission
GOSIP	Government OSI Profile. (USA and UK)
GPIB	General Purpose Interface Bus (= IEEE-488)
GUI	Graphical User Interface
GUIDE	Guidance for Users of Integrated Dp Equipment
HDLC	High-Level Data Link Control (ISO)
HLLAPI	High-Level Language Application Programming Interface
HONE	Hands-On Network Environment (IBM)
HPFS	High-Performance File System (IBM/Microsoft)
IBM	International Business Machines Corporation
IBU	Independent Business Unit
IC	Integrated Circuit
ICA	Integrated Communications Adapter (IBM)
ICI	Interface Control Information (ISO)
ICLAS	IBM Classroom LAN Administration System (IBM)
ICMP	Internet Control Message Protocol (TCP/IP)
IDAP	IBM Developer Assistance Program
IDE	Integrated Device Electronics
IDI	Initial Domain Identifier (ISO)
IDU	Interface Data Unit (ISO)
IEC	International Electrotechnical Commission

IEEE	Institute of Electrical and Electronics Engineers (USA)
IFIP	International Federation for Information Processing
IFS	Installable File System
IOV	Independent Option Vendor
IP	Internet Protocol (ISO and TCP/IP)
IPABX	ISDN Private Automatic Branch EXchange
IPC	Inter-Process Communications
IPDS	Intelligent Printer Data Stream (IBM)
IPM	Images Per Minute
IML	Initial Microcode Load
IPL	Initial Program Load
IPX	Internetwork Protocol eXchange (Novell)
IR	Industry Remarketer
IS	International Standard
ISA	Industry Standard Architecture
ISAM	Indexed Sequential Access Method
ISDN	Integrated Services Digital Network
ISO	International Standards Organization
ISV	Independent Software Vendor
I/O	Input/Output (E/A = Ein-/Ausgabe)
JES-2	Job Entry Subsystem 2 (IBM)
JMT	Job Management and Transfer (ISO)
Kb	Kilobit (= 1,024 bits)
KB	Kilobyte (= 1,024 bytes)
KVA	KiloVolt Amperes
LAPB	Link Access Procedure - Balanced (OSI)
LAPM	Link Access Procedure - Modem (OSI)
LAN	Local Area Network
LEN	Low-Entry Networking
LIM EMS	Lotus/Intel/Microsoft Expanded Memory Specification
LLC	Logical Link Control
LPM	Lines Per Minute
LRU	Least Recently Used
LSB	Least Significant Bit
LSI	Large-Scale Integration
LU	Logical Unit (IBM)
LUA	Logical Unit Address
LU2	Logical Unit 2 (cluster controller)
LU6.2	Logical Unit 6.2 (= APPC)
M&ED	Multimedia and Education Division (IBM)
M&S	Marketing and Services Division (IBM)
MAC	Media Access Control
MAN	Metropolitan Area Network
MAP	Manufacturing Automation Protocol (General Motors)
MAPICS	Management Accounting, Production, and Information Control System
MAU	Multistation Access Unit (IBM)
MAU	Medium Attachment Unit (802.3)
Mb	Megabit (= 1,024 Kb = 1,048,560 bits)
MB	Megabyte (= 1,024 KB = 1,048,560 bytes)
MCA	Trademark of Music Corporation of America (but often used, incorrectly, to signify Micro Channel Architecture)

MCDA	Micro Channel Developers Association
MCGA	Multi-Color Graphics Array (IBM)
Megapel	1024x1024 resolution (or higher)
MFI	MainFrame Interactive terminal (IBM)
MFLOPS	Millions of FLoating-point OPerations per Second
MFM	Modified Frequency Modulation
MHS	Message Handling System (ISO)
MICR	Magnetic Ink Character Reader
MIPS	Millions of Instructions Per Second
MIS	Management Information System
MNP	Microcom Networking Protocol
MODCA	Mixed-Object Document Content Architecture
modem	MOdulation/DEModulation
MOTIS	Message Oriented Text Interchange System (ISO)
MPS	Multiple Port Sharing (IBM)
MSB	Most Significant Bit
MTA	Message Transfer Agent (X.400)
MTBF	Mean Time Between Failures
MTS	Message Transfer Service
MTTR	Mean Time To Repair
MVS	Multiple Virtual Storage (IBM)
MVS/XA	Multiple Virtual Storage/eXtended Architecture (IBM)
NBS	National Bureau of Standards (= NIST)
NCP	Network Control Program (IBM)
NDD	National Distribution Division (IBM)
NDIS	Network Driver Interface Specification (3Com/Microsoft)
NDM	Network Data Mover (IBM)
NetBIOS	Network Basic Input/Output System (IBM)
NIST	National Institute for Standards and Technology (= NBS)
NIU	Network Interface Unit
NJE	Network Job Entry (IBM)
NLDM	Network Logical Data Management (IBM)
NMPF	Network Management Productivity Facility (IBM)
NMR	Network Management Reporter (IBM)
NOS	Network Operating System
NPDA	Network Problem Determination Application (IBM)
NPM	Network Performance Monitor (IBM)
NPSI	NCP Packet Switching Interface (IBM)
NRZ	Non-Return to Zero (IBM)
NRZI	Non-Return to Zero Inverted (IBM)
NSAP	Network Service Access Point (OSI)
NSD	National Service Division (IBM)
nsec	Nanosekunden
NSP	Name Service Protocol (TCP/IP)
NSTL	National Software Testing Laboratory
ODA	Office Document Architecture
ODIF	Office Document Interchange Format
OEM	Other Equipment Manufacturer or Original Equipment Manufacturer)
OLTP	OnLine Transaction Processing
OOP	Object-Oriented Programming
OOPS	Object-Oriented Programming System
OS/2	Operating System/2 (IBM/Microsoft)

OS/2 SE	Operating System/2 Standard Edition (IBM)
OS/2 EE	Operating System/2 Extended Edition (IBM)
OSF	Open Systems Foundation
OSF/1	OSF operating system based on UNIX/AIX
OSI	Open Systems Interconnection (ISO)
OSNS	OSI Subsystem Network Services (IBM)
OSTS	OSI Subsystem Transport Services (IBM)
PABX	Private Automatic Branch Exchange
PAD	Packet Assembler/Disassembler
PARC	Palo Alto Research Center (Xerox)
PASF	PROFS Applications Support Feature (IBM)
PASR	Problem Analysis and Status Report (IBM)
PDS	Personal Decision Series (IBM)
PL/I	Programming Language 1
PM	Presentation Manager (Microsoft/IBM)
PM/X	Presentation Manager/X (Microsoft/HP)
POWER	Performance Optimization With Enhanced RISC (IBM)
PPM	Pages Per Minute
PROFS	PRofessional OFfice System (IBM)
PROM	Programmable Read-Only Memory
PS/1	Personal System/1 (IBM)
PS/2	Personal System/2 (IBM)
PSDN	Packet-Switched Data Network
PSAP	Presentation Service Access Point
PTF	Program Temporary Fix
PTR	Program Trouble Report
PTS	Personal Typing System (IBM)
PU	Physical Unit (IBM)
QM	Query Manager (IBM OS.2 EE)
QMF	Query Management Facility (IBM)
RAM	Random Access Memory
RARP	Reverse Address Resolution Protocol (TCP/IP)
RDA	Remote Database Access
RDS	Remote Data Services (OS/2 EE)
REXX	REstructured eXtended eXecutor procedure language
RFC	Request For Comment (DARPA)
RFTDCA	Revised Form Text Document Content Architecture
RIP	Routing Information Protocol (TCP/IP)
RIPL	Remote Initial Program Load
RISC	Reduced Instruction Set Computer
RJE	Remote Job Entry
RLE	Run Length Encoding
RLL	Run Length Limited
ROECE	IBM Regional Office for Europe, Central and East
ROM	Read Only Memory
ROS	Read Only Storage
ROSE	Remote Operations Service Element (OSI)
RPC	Remote Procedure Call
RPG II	Report Program Generator (language) II
RPL	Remote Program Load
RPQ	Request for Price Quotation
RS/6000	RISC System/6000 (IBM)
RT/CMOS	Reat-Time-Clock/Complementary Metal Oxide Semiconductor
RTS	Request To Send

SAA	System Application Architecture (IBM)
SAP	Service Access Point (ISO)
SCSI	Small Computer Systems Interface (ANSI)
SDLC	Synchronous Data Link Control (IBM)
SDK	Software Developers Kit (Microsoft)
SGMP	Simple Gateway Management Protocol (TCP/IP)
SHARE	Society to Help Alleviate Redundant Effort
SHM	Short Hold Mode (IBM)
SIMM	Single Inline Memory Module
SIP	Single Inline Package
SLR	Service Level Reporter (IBM)
SLSS	System Library Subscription Service (IBM)
SMK	Software Migration Kit (Microsoft)
SMT	Surface-Mount Technology
SMTP	Simple Mail Transfer Protocol (TCP/IP)
SNA	System Network Architecture (IBM)
SNADS	SNA Distribution Services (IBM)
SNAPDU	SNA Protocol Data Unit (IBM)
SNI	SNA Interconnect (IBM)
SNMP	Simple Network Management Protocol, old name, now TCP/IP
SNMP	Systems Network Management Protocol (TCP/IP)
SQL	Structured Query Language
SQL/DS	Structured Query Language/Data System
SRAM	Static Random Access Memory
SRI	Stanford Research Institute
SRPI	Server Requester Programming Interface (IBM)
SSG	Service Support Guide (IBM)
SYTOS	SYtron Tape Operating System
Tb	Terabit (= 1,024Gb = 1,048,560Mb)
TB	Terabyte (= 1,024GB = 1,048,560MB)
TCF	Transparent Computing Facility (IBM)
TCP/IP	Transmission Control Protocol/Internet Protocol
TIC	Token-Ring Interface Coupler (IBM)
TMN	Telecommunications Management Network
TMS	Telecommunications Management System
TOP	Technical and Office Protocol
TOS	Tape Operating System
TRN	Token-Ring Network
TSR	Terminate and stay resident
UDP	User Datagram Protocol (TCP/IP)
U/L	Underwriter's Labratory (USA)
UNIX	An operating system developed by AT&T
VAN	Value-Added Network
VDT	Video Display Terminal
VDU	Video Display Unit
VGA	Video Graphics Array (IBM)
VLSI	Very Large-Scale Integration
VM	Virtual Machine (IBM)
VRAM	Video Random Access Memory
VT	Virtual Terminal (ISO)
VT-100	Terminal protocol (DEC)
VTAM	Virtual Telecommunications Access Method (IBM)

WAN	Wide Area Network
WORM	Write-Once, Read-Many
WYSIWYG	What You See Is What You Get
X.121	Packet Addressing (CCITT)
X.25	Packet Switching (CCITT)
X.400	Message Handling (CCITT)
X.500	The Directory Service (Address service) (CCITT)
XI	X.25 SNA Interconnect (IBM)
XMA	EXpanded Memory Adapter (IBM)
4GL	Fourth Generation Language
10Base-2	Ethernet thin-cable connection
10Base-5	Ethernet thick-cable connection
10Base-T	Ethernet twisted pair connection

Anhang C. Index

Numerische Stichwörter

4004 3-4
8008 3-4
80286 3-4
80287 3-4
80386 3-4, 3-9
80386DX 3-4
80386SLC 3-12
80386SX 3-4, 3-12
80387 3-4, 3-13
80486 3-4, 3-14
80486DX2 3-18
80486SX 3-4, 3-15
80487 3-4, 3-15
8086 3-4
8087 3-4
8088 3-4
80x86 / 80x87, Prozessorfamilie 3-8

A

Änderungsstand 7-29
Anlegen und Löschen von Dateien 8-14
Anlegen von Dateien aus Programmen heraus 8-15
Anlegen von Dateien mit Schablonen 8-14
Antwort-Datei 2-22
Anwendungsprofil für welche Hardware Sie sich entscheiden sollten. 3-31
Anzeige des Änderungsstandes des OS/2 7-29
anzeigen Bilddateien 10-16
Arbeitsoberfläche 1-12, 1-13
Arbeitsoberfläche, Einbinden von DOS-Programmen 6-8
Arbeitsordner 1-10
Architektur, Computer- 3-19

Architektur, Mikrokanal 3-20
Arithmetic Logical Unit (ALU) 3-6
ATM, Prinzip des 10-4
AUDIO_ADAPTER_SHARING 6-19
Aufteilung der Festplatte(n) 2-4

B

BAT-Dateien unter OS/2 9-5
Befehle für Stapeldateien 9-6
Begriffe, neue 1-10
Behälter 1-10
Behandlung der DOS-Einstellungen 6-15
Betriebssysteme, Installation anderer 2-19
Bilddateien anzeigen 10-16
Blocks 7-8
Boot Manager 1-6
Boot-Manager 2-9, 2-10, 2-14
Booten von DOS-Betriebssystemen in Virtuellen Maschinen 6-21
Booten von virtuellen DOS-Maschinen 6-21
BUFFERS 7-9
BUS 3-6
Bus Interface Unit (BIU) 3-6

C

Cache 3-12, 3-14, 7-8
CAD/CAM 3-31
CMD-Dateien unter OS/2 9-4
COM_HOLD 6-19
COM_SELECT 6-20
Computer-Architektur 3-19
CONFIG.SYS 7-7
Container 1-10
Coprozessor, mathematischer 3-12, 3-14
Corrective Service Diskettes 7-28
CREATEDD 7-20
CSD 7-28

D

Datei-Manager 1-14
Dateien, Anlegen mit Schablonen 8-14
Dateien, Anlegen und Löschen von 8-14
Datenaustausch zwischen OS/2-Programmen und Windows-Programmen 6-28
Datenbestände, Löschen 8-16
Datenbestände, Sichern 8-22
Desktop 1-12, 1-13
Diagnose 7-16
Dialog Editor 11-12
Dienstprogramme TRACE und TRACEFMT 7-23
Direktmanipulation 1-13
DISKCACHE 7-8
DOS Protected Mode Interface 6-13
DOS- und OS/2 Stapeldateien 9-4
DOS-Betriebssystem, Virtuelles Booten 6-21
DOS-Einstellungen 6-14
DOS-Einstellungen, Behandlung von 6-15
DOS-Maschinen, Einrichten der Virtuellen 6-7
DOS-Maschinen, Virtuelle 6-5
DOS-Programme, Einbinden in die Arbeitsoberfläche 6-8
DOS_AUTOEXEC 6-15
DOS_BACKGROUND_EXECUTION 6-15
DOS_BREAK 6-15
DOS_DEVICE 6-19
DOS_FCBS 6-16
DOS_FCBS_KEEP 6-16
DOS_FILES 6-16
DOS_HIGH 6-16
DOS_LASTDRIVE 6-15
DOS_RMSIZE 6-16
DOS_SHELL 6-15
DOS_STARTUP_DRIVE 6-20
DOS_UMB 6-16

DOS_VERSION 6-15
DPMI_DOS_API 6-17
DPMI_MEMORY_LIMIT 6-17
DPMI_NETWORK_BUFF_SIZE 6-17
Drag and drop 1-13
Drives 1-14
Druck-Manager 1-13
Druckausgabe, Konzept der 5-4
Drucken im LAN 5-25
Drucker Objekte 1-13
Drucker-Objekt 1-10
Druckerobjekt, Kontext-Menü 5-6
Druckerobjekt, Kopieren 5-10
DUAL-BOOT 2-19
DUMP 7-20
Dynamischer Datenaustausch 6-28

E

Echte Namen und Titel 8-12
Editor EPM, Erweiterter 11-3
Editor, Dialog 11-12
Editor, Font 11-10
Einbinden von DOS-Programmen in die Arbeitsoberfläche 6-8
Einheiten-Kontext 10-15
Einrichten der Virtuellen DOS-Maschinen 6-7
Einrichten der Windows-Umgebung 6-27
Einsatz von Stapeldateien 9-3
EMS_FRAME_LOCATION 6-17
EMS_HIGH_OS_MAP_REGION 6-17
EMS_LOW_OS_MAP_REGION 6-17
EMS_MEMORY_LIMIT 6-17
Entwicklung der Mikroprozessoren 3-4
EPM, Erweiterter Editor 11-3
Erweiterter Editor EPM 11-3
Erweiterter Modus 2-14
Execution Unit (EU) 3-6

Expanded Memory Storage 6-12
Extended Memory Specification 6-13

F
FDISK 2-13
Festplatte 3-31
Festplatte(n), Aufteilung der 2-4
File System, High Performance 8-7
Folder 1-10, 1-13
Font Editor 11-10
Funktionen eines Druckerobjektes 5-5
Funktionen, geänderte 1-13

G
Geänderte Funktionen 1-13
Gemeinsamer Speicher 4-11
Grafik-Segmente 10-14
Grafik-Segmente und Metafiles 10-14
Grafikformate 10-17
Grafikkarte 3-31
Grafische Grundfunktionen 10-8
Graphic primitives 10-8

H
Hauptspeicher 3-31
High Memory Area 6-11
High Performance File System 8-7
HW_NOSOUND 6-19
HW_ROM_TO_RAM 6-19
HW_TIMER 6-19

I
I/O-Adaptern 3-6
I/O-Kanal 3-6
i486 3-4, 3-14
i487 3-4

IDLE_SECONDS 6-19
IDLE_SENSITIVITY 6-19
Informations-Darstellung 11-15
Informations-Entwicklung 11-13
Installation anderer Betriebssysteme 2-19
Installation von Windows-Programmen 6-31
Installationsplanung A-1
Installieren eines zusätzlichen Druckerobjektes 5-10
Installieren von Schriftarten 10-5
INT_DURING_IO 6-20
Intel 80x86 / 80x87, Prozessorfamilie 3-8
IOPL 4-6
IPF-Datei, Schreiben einer 11-16
IPF-Tag-Übersicht 11-26

K

KBD_ALTHOME_BYPASS 6-20
KBD_BUFFER_EXTEND 6-20
KBD_CTRL_BYPASS 6-20
KBD_RATE_LOCK 6-20
Konzept der Druckausgabe 5-4
Koordinatensysteme 10-10
Kopieren eines Druckerobjektes 5-10

L

Laufwerke 1-14, 7-17
Laufwerke, Logische 2-16
Logische Laufwerke 1-14, 2-16
Logischer Drucker 1-13
logisches Laufwerk 2-4
Löschen von Dateien, Anlegen und 8-14
Löschen von Datenbeständen 8-16

M

mathematischer Coprozessor 3-12, 3-14
MAXWAIT 7-7

Megahertz 3-18
MEM_EXCLUDE_REGIONS 6-17
MEM_INCLUDE_REGIONS 6-17
MEMMAN 7-8
Memory Overcommitment 4-14
Metafiles 10-14
MHz 3-18
Microchannel (MC) 3-19
Migration von Programmen nach OS/2 PM 6-4
Mikrokanal 3-19
Mikrokanal Architektur 3-20
Mikroprozessoren, Entwicklung der 3-4
minfree 7-8
Model, workplace 1-10
MOUSE_EXCLUSIVE_ACCESS 6-15
Multitasking 3-31, 4-17
Multithreading 4-18

N

Namen und Titel, Echte 8-12
Neue Begriffe 1-10
Notebook 1-10
Notizbuch 1-10

O

Object 1-10
Objects, printer 1-13
Objekt 1-10
Ordner 1-10, 1-13
OS/2 PM, Migration von Programmen nach 6-4
OS/2- und DOS Stapeldateien 9-4
OS/2-Programme und Windows-Programme, Datenaustausch zwischen 6-28
OS2.INI 7-3
OS2SYS.INI 7-3

P

Paging 4-14
Papierkorb 1-10
Partition 2-4
PATCH 7-28
Pentium 3-4, 3-16
PICVIEW 10-16
Planung der Installation 2-6
Plattenorganisation 8-5
PM Chart 10-17
PM-Programme, Struktur von 11-29
PM, Migration von Programmen nach OS/2 6-4
Präsentations Manager 1-7
Primäre Partition 2-4
PRINT_TIMEOUT 6-19
Printer Object 1-10
Printer objects 1-13
Prinzip des ATM 10-4
PRIORITY 7-7
Privileged Level 4-6
Programme, Unterstützende 11-35
Programmieren mit SOM 11-32
Programming Tools and Information 11-9
Protected Modus 3-8
Prozesse 4-20
Prozessor 3-6, 3-31
Prozessorcache 3-12, 3-14
Prozessorfamilie Intel 80x86 / 80x87 3-8
PSTAT 7-27

R

RAM 3-6
REAL Modus 3-8
Register 3-7
Remote Installation 2-23
response file 2-22
ROM 3-6

S

Schablonen, Anlegen von Dateien mit 8-14
Schriftarten unter OS/2 10-3
Schriftarten, Installieren von 10-5
Schutzring 4-6
Separator pages 5-17
setboot 2-13
Shared Memory 4-11
Shredder 1-10
Sichern von Datenbeständen 8-22
Singletasking 4-17
Sitzung 4-18
SOM, Programmieren mit 11-32
Speicherauszug 7-20
Speichermanagement unter DOS und WINDOWS 6-9
Speichermodell 4-8
Stapeldatei, Anwendungsbeispiel 9-3
Stapeldateien, Befehle für 9-6
Stapeldateien, Einsatz von 9-3
Stapeldateien, Tabelle mit Befehlen für 9-6
STARTUP-Datei 9-9
Struktur von PM-Programmen 11-29
Swap 7-8
SWAPPATH 7-8
Swapping 4-14
SYSLEVEL 7-29
System Setup 7-3
Systemanpassung 7-1
Systemkonfiguration 7-3
Systemoptimierung 7-7
Systemplatine 3-6
Systemstart 9-8

T

Taktrate 3-18, 3-31
THREADS 7-8, 7-27

TIMESLICE 7-7
Tips und Hinweise zu OS/2 Version 2 7-30
Titel, Echte Namen und 8-12
TRACE und TRACEFMT, Dienstprogramme 7-23
TRACEFMT, Dienstprogramm 7-23
TRAPDUMP 7-20
Trennseiten 5-17

U

UNDELETE 8-17
Unterstützende Programme 11-35
Upper Memory Blocks 6-10

V

Value Set 1-11
VIDEO_8514A_XGA_IOTRAP 6-18
VIDEO_FASTPASTE 6-18
VIDEO_MODE_RESTRICTION 6-18
VIDEO_ONDEMAND_MEMORY 6-18
VIDEO_RETRACE_EMULATION 6-18
VIDEO_ROM_EMULATION 6-18
VIDEO_SWITCH_NOTIFICATION 6-18
VIDEO_WINDOW_REFRESH 6-19
Virtuelle DOS-Maschinen 6-5
Virtuellen DOS-Maschinen, Einrichten der 6-7
Virtueller Speicher 4-14
VMB, Hinweise zur Arbeit mit dem 6-21

W

Warteschlangen 1-12
Wege des Druckjobs 5-21
WIN_CLIPBOARD 6-20
WIN_DDE 6-20
WIN_RUNMODE 6-20
Windows-Programme und OS/2-Programme Datenaustausch zwischen 6-28

Windows-Programme unter OS/2 Version 2 6-27
Windows-Programme, Installation von 6-31
Windows-Umgebung, Einrichten der 6-27
Work Area 1-10
Workplace model 1-10
Workplace shell 1-10

X

XMS_MEMORY_LIMIT 6-17
XMS_MINIMUM_HMA 6-17

Z

Zwischenablage 6-28